MASTER EN SCIENCES INFORMATIQUES

Une approche transformationnelle pour la conception de tableurs multidimentionnels pour l'usage domestique

Jadoul, Laurent

Award date:
2017

Awarding institution:
Universite de Namur

UNIVERSITÉ DE NAMUR
Faculté d'informatique
Année académique 2016–2017

Une approche transformationnelle pour la conception de tableurs multidimentionnels pour l'usage domestique

Laurent JADOUL

Promoteur : _____ (Signature pour approbation du dépôt - REE art. 40)
Michaël PETIT

Mémoire présenté en vue de l'obtention du grade de
Master en Sciences Informatiques.

Je tiens à remercier mon promoteur, le professeur Michaël Petit, pour son aide et ses conseils durant la réalisation de ce mémoire. Je remercie également mon époux Pascal, ainsi que ma famille pour tout le soutien qu'ils m'ont apporté durant ces années d'étude. Enfin, je souhaite remercier l'ensemble des membres de la faculté d'informatique de l'université de Namur ainsi que mes collègues et amis.

Résumé

Ce mémoire propose une méthode de résolution d'un problème de calcul dimensionné à l'attention d'un utilisateur peu habitué aux techniques de modélisation de données. La méthode proposée va combiner la modélisation dimensionnelle avec la modélisation Hainaut d'un problème de calcul pour ensuite réaliser une implémentation dans le tableur Excel grâce à l'outil PowerPivot et le langage DAX. Le résultat sera la création d'un entrepôt de données dans un classeur Excel.

Cette méthode va permettre d'aider l'utilisateur à structurer les données en sa possession et lui permettre d'ajouter des calculs sur les données afin de répondre aux questions qu'il se pose.

Mots-clés

Modèle de données, modélisation dimensionnelle, modélisation d'un problème de calcul dimensionné, entrepôt de données, faits, mesures, dimensions, grandeurs dimensionnées, Excel, Powerpivot, DAX, tableaux croisés dynamiques.

Table des matières

Introduction

Comment faire quand on est indépendant, patron de petite entreprise ou bien simple particulier et que l'on souhaite analyser des données dimensionnées ? Comment faire quand on ne souhaite pas investir une grande partie de son budget dans des solutions informatiques d'analyses décisionnelles ? Finalement, comment réaliser cette analyse alors qu'on ne dispose pas de compétences informatiques en modélisation de données ?

Il est courant de vouloir suivre la trajectoire budgétaire de son entreprise ou bien de souhaiter établir des scénarios afin de voir où les investissements seraient les plus rentables. La plupart du temps, les personnes se tournent vers un tableur tel qu'Excel de chez Microsoft® pour introduire les données et ensuite créer les formules permettant d'obtenir un résultat. Comment faire par contre pour représenter des données dimensionnées dans un tableur et ensuite créer les calculs qui répondront aux interrogations des utilisateurs ?

L'objectif de ce mémoire sera de trouver une méthodologie permettant de résoudre des problèmes de calculs sur des données dimensionnées dans le cadre d'un usage domestique.

La méthode qui sera développée va permettre à un public n'ayant pas de grande expérience informatique de mettre en place un petit entrepôt de données dans un classeur Excel 2016 afin de pouvoir analyser ces données et répondre à des problèmes de calcul. Cette méthode représentera le résultat de la fusion de deux modèles en une solution pratique utilisant des outils complémentaires à Excel.

Ces deux modèles sont d'une part un modèle permettant de créer un entrepôt de données et d'autre part un modèle de résolution de problème de calcul dont les données sont dimensionnées.

La première partie de ce mémoire expliquera les concepts existants dans chaque modélisation ainsi que les techniques à utiliser pour concevoir les deux modèles. Nous aborderons ensuite les solutions techniques existantes permettant de résoudre des problèmes de calculs sur des données dimensionnées,

en nous concentrant plus particulièrement sur le tableur Excel 2016 de chez Microsoft®, l'outil PowerPivot et le langage DAX. Dans la seconde partie du mémoire, nous verrons comment combiner les deux modélisations abordées dans la première partie en un seul modèle. C'est ensuite que nous décrirons les étapes nécessaires à la transformation du modèle de problème de calcul en une implémentation dans le tableur Excel.

Première partie

État de l'art

Chapitre 1

Modéliser un problème à plusieurs dimensions

1.1 Introduction

Résoudre un problème de calcul peu complexe est facile : quelques données, des règles de calcul simples et tout utilisateur d'un tableur pourra implémenter une solution rapidement tout en minimisant les erreurs. En revanche, dès que le problème se complexifie, la probabilité qu'une ou plusieurs erreurs surviennent augmente et le risque et le risque d'obtenir des résultats erronés sera d'autant plus grand qu'ils seront difficiles à détecter. Il est donc nécessaire de passer par une étape de conceptualisation de la problématique afin de valider et renforcer la solution qui sera développée.

Dans ce chapitre, nous aborderons deux types de modélisations : la modélisation dimensionnelle pour la réalisation d'un entrepôt de données et la modélisation d'un problème de calcul développé par Jean-Luc HAINAUT. Ces deux types de modèles vont permettre, en les combinant, de transformer un problème de calcul en une solution dans un tableur domestique tel que Microsoft® Excel.

Afin de mieux illustrer certains concepts, un exemple simple va nous suivre tout au long de ce mémoire. L'exemple choisi est celui d'une entreprise de vente de vélos et d'accessoires auprès de clients à travers le monde [1]. Cet exemple est basé sur le cas d'étude développé par Microsoft® tel qu'utilisé durant leurs formations à l'informatique décisionnelle, aussi appelé Business Intelligence.

1. https://technet.microsoft.com/fr-fr/library/ms124825(v=sql.100).aspx

1.2 Quelques mots sur la Business Intelligence

Avant d'aborder les modélisations nous allons parler de l'informatique décisionnelle aussi nommée **business intelligence** (BI). Ce terme désigne une partie de l'informatique à destination des dirigeants d'entreprises et des analystes. Son but est d'offrir une vue d'ensemble des activités de l'entreprise. Derrière ce terme se retrouvent aussi bien des moyens, des outils ou des méthodes qui servent à collecter, analyser et présenter des données dans le but d'offrir aux utilisateurs un outil d'aide à la décision.

Principalement théorisée par Ralph KIMBALL, la business intelligence repose sur une architecture qui se découpe en trois parties :

- la *collecte* périodique des données qui peuvent provenir de sources hétérogènes et qui sont stockées dans un **entrepôt de données** ;
- l'*analyse* qui va permettre de restructurer les données et éventuellement de les enrichir en les combinant soit à d'autres données résultant d'agrégations, soit à des calculs sur les données d'entrée ;
- la *présentation* des données qui peut se décliner sous différentes formes telles que des rapports, des tableaux de bords ou bien des applications.

Dans [Kimball and Ross, 2013] nous pouvons trouver des exigences pour la réalisation d'un système BI/entrepôt de données qui découlent de l'expérience de Ralph KIMBALL :

- l'information doit être rendue facilement accessible pour l'utilisateur final. La structure des données et les libellés doivent ainsi être le reflet de la pensée de l'utilisateur ;
- l'information doit être cohérente. Les données présentes dans un entrepôt de données viennent souvent de plusieurs sources, il est donc important de vérifier leur qualité mais également la cohérence de celles-ci. Cela implique d'avoir des libellés et des définitions des données qui restent communes à travers le système ;
- le système doit s'adapter aux changements. Il faut donc veiller à ne pas causer de rupture dans l'accès aux données déjà présentes dans l'entrepôt de données lorsque l'utilisateur formule de nouvelles exigences ;
- le système doit protéger les informations. Un entrepôt de données contient beaucoup d'informations sur la vie de l'entreprise, il est donc nécessaire de veiller à le sécuriser et à n'autoriser l'accès aux informations qu'aux personnes habilitées ;

- le système doit être digne de confiance afin d'améliorer la prise de décision ;
- le système doit être pleinement accepté par les utilisateurs pour être un succès dans la prise de décisions.

L'informatique décisionnelle est donc un système qui va permettre d'analyser et d'interpréter les données dans un but de reporting auprès des décideurs d'entreprises. En chargeant et transformant les données brutes dans des entrepôts de données, tout en proposant des outils permettant de réaliser des rapports sur ces données, la Business Intelligence va permettre de suivre des indicateurs stratégiques pour l'entreprise.

1.3 La modélisation dimensionnelle

La réalisation d'un entrepôt de données passe par la création d'un **modèle dimensionnel** [Kimball and Ross, 2013] ; [Malinowski and Zimányi, 2008] ; [Burquier, 2007]. La modélisation dimensionnelle est une technique de conception logique qui consiste à mettre en relation des **faits** et des **dimensions**. Elle va permettre de structurer les données de manière à les rendre intuitives aux utilisateurs et d'offrir de bonnes performances aux requêtes. Si le terme "dimensionnel" semble être complexe, celui-ci correspond pourtant à une approche qui se veut naturelle et qui décrit l'activité de l'entreprise.

Il n'existe pas à ce jour un seul modèle dimensionnel qui fasse l'unanimité au sein de la communauté scientifique mais bien plusieurs modèles avec leurs notations propres et leurs techniques de modélisation.
Dans [Malinowski and Zimányi, 2008], une méthode de modélisation semblable à la modélisation d'un système de bases de données relationnelles est proposée. Cette méthode propose de réaliser un modèle conceptuel, ensuite un modèle logique et enfin un modèle physique de l'entrepôt de données à créer. Il est également proposé une notation spécifique pour la modélisation conceptuelle qui est nommée **MultiDim**. Cette notation, dont certains éléments sont visibles dans la figure 1.1, est une notation parmi d'autres telle que la notation **ADAPT** (Application Design for Analytical Processing Technologies) développée par Symmetry Corp ou bien la notation **Gold** utilisant les bases de la notation UML.

À la différence du modèle entité-relation (ER) qui représente les données sous la forme d'entités et de relations, le modèle dimensionnel représente celles-ci comme des faits et des dimensions. Un des avantages d'un modèle dimen-

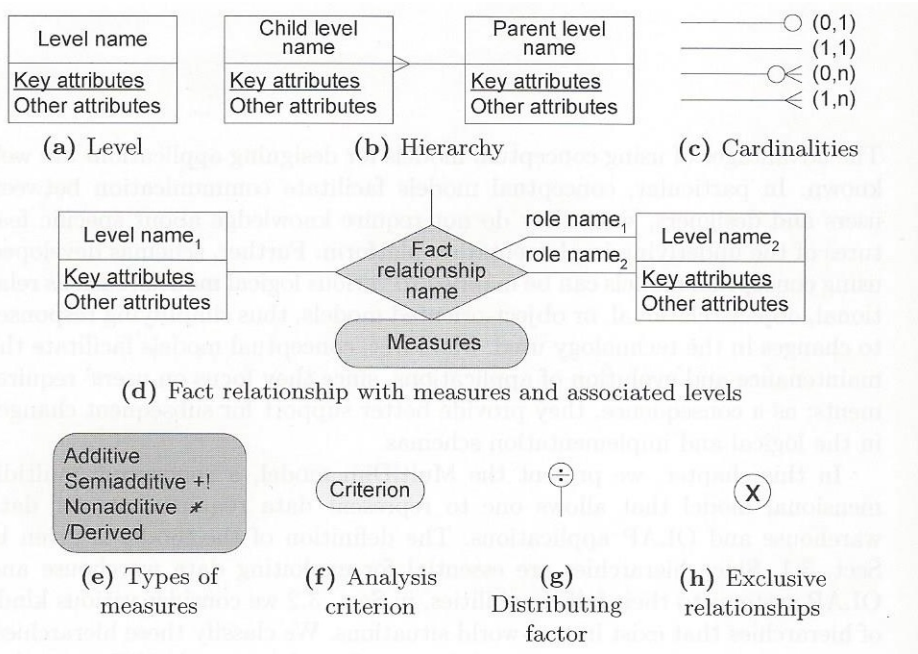


FIGURE 1.1 – Notation du modèle MultiDim repris dans [Malinowski and Zimányi, 2008]

sionnel concerne la compréhension du modèle par les utilisateurs en regroupant les données en catégories "naturelles" pour celui-ci. Un autre avantage concerne la performance d'accès aux données. Le modèle dimensionnel peut ne pas être normalisé comme le modèle entité-relation, ce qui évite lors de l'implémentation du modèle dimensionnel des jointures coûteuses en temps d'accès.

Dans cette section du chapitre nous allons aborder les concepts théoriques de la modélisation dimensionnelle d'après [Kimball and Ross, 2013]. La conception d'un tel modèle sera développé plus en détail dans le chapitre 2 de ce mémoire.

1.3.1 Les approches lors de la construction d'un modèle dimensionnel

Lors de la construction d'un entrepôt de données, plusieurs approches de modélisation sont possibles. Un système d'informatique décisionnelle peut être composé d'un entrepôt de données volumineux et contenant beaucoup de di-

mensions et de tables de faits différentes. La plupart du temps, un entrepôt de données dans une entreprise est composé de plusieurs **Data Mart** (magasins de données) qui sont chacun spécifiques à une partie métier de l'entreprise.

Ces magasins de données vont parfois être amenés à partager des dimensions communes. Le choix de l'approche de construction est donc importante afin d'éviter la création de dimensions ou de tables de faits redondantes.

Approche Top-Down

L'approche Top-Down, également nommée approche descendante, consiste à concevoir l'entrepôt de données intégralement dès le début de la mise en place du système d'informatique décisionnelle. Elle nécessite de connaitre à l'avance toutes les dimensions et tous les faits afin de livrer une solution basée sur des méthodes et technologies éprouvées des bases de données. Cette approche théorisée par William H. INMON permet d'offrir une architecture complète et intégrée, réutilisant les données, évitant les redondances et proposant une vision claire des données de l'entreprise et du travail à réaliser.

Parmi les inconvénients de cette approche, citons le fait que celle-ci fait appel à une méthodologie lourde, contraignante et nécessitant beaucoup de temps. C'est cette approche qui est utilisée dans [Malinowski and Zimányi, 2008] afin de réaliser un entrepôt de données.

Approche Bottom-Up

L'approche Bottom-Up (approche ascendante) est l'opposé de l'approche précédente. Elle consiste à concevoir les magasins de données un à un séparément pour ensuite les regrouper dans un seul entrepôt de données. L'objectif étant de livrer une solution simple à réaliser, rapide et plus orientée vers les utilisateurs.

Les inconvénients de l'approche sont le volume important de travail nécessaire pour rassembler les différents magasins de données dans un seul entrepôt, le risque de redondance et la moindre efficacité à long terme en cas d'évolution de l'entrepôt de données.

1.3.2 Les tables de faits

Les faits sont ce sur quoi va porter l'analyse. Les tables de faits contiennent des données appelées **mesures** qui relatent la vie de l'entreprise. Dans notre

exemple, la société vend des produits auprès de revendeurs et de particuliers. Ces ventes sont enregistrées dans le système informatique de la société. On peut considérer comme un **fait** une ligne d'une commande et dont une des **mesures** serait la quantité de l'article commandé.

Une table de faits d'un entrepôt de données correspond, dans un modèle dimensionnel, à un seul **processus métier** de l'entreprise. La table de faits collecte des mesures générées par les événements de ce processus. Les thématiques des tables de faits peuvent porter sur les ventes (quantités vendues, montants facturés, coûts de livraisons,...), sur les ressources humaines (absentéisme, nombre d'accidents, heures travaillées,...) ou bien sur les stocks.

Les tables de faits d'un entrepôt de données sont les tables qui contiendront le plus de lignes d'enregistrements. Ces tables sont typiquement structurées en deux types de colonnes : Les colonnes de type **clés étrangères** qui viennent faire la jointure entre les faits et les dimensions, et les colonnes de type **mesures** qui représentent les valeurs numériques d'un fait. Dans l'exemple de la figure 1.2, on peut remarquer que la table de faits "Vente" contient en premier lieu les clés étrangères des dimensions et ensuite les mesures qui ont été définies pour décrire un fait ayant un rapport avec une vente.

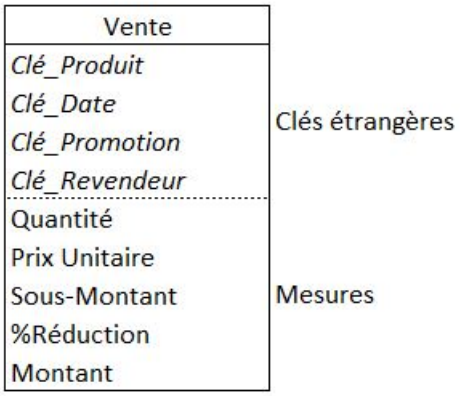


FIGURE 1.2 – Exemple de structure d'une table de faits

Parmi les mesures, nous pouvons observer deux sortes de mesures : des mesures que je nommerai **factuelles** et des mesures dites **calculées**. Les mesures factuelles représentent des valeurs qui ont été introduites dans le système suite à un événement du processus métier en rapport avec la table de faits. À l'opposé, une mesure calculée est le résultat d'une opération basé sur des mesures factuelle et/ou calculées. Dans l'exemple repris dans la figure 1.2,

la quantité, le prix unitaire et le pourcentage de réduction sont des mesures factuelles. Elles correspondent à une vente réalisée par l'entreprise et ces valeurs ont été encodées dans le système au moment où l'événement s'est produit. En revanche, les mesures *sous-montant* et *montant* sont des mesures calculées car leurs valeurs sont le résultat d'un calcul.

Nous verrons plus tard, lors de l'implémentation de la solution, que certaines mesures calculées vont être évaluées suivant plusieurs contextes. En fonction du contexte, la mesure créée sera soit une **mesure calculée** ou bien une **colonne calculée**.

La granularité

Un fait représente donc un événement de la vie de l'entreprise dans une table de faits. Ces événements doivent être définis grâce à une granularité ou **grain**. Le grain établit clairement ce qu'un simple fait représente. Ce grain doit être choisi avant même de définir les dimensions et les faits, car ceux-ci devront obligatoirement être cohérents avec la granularité choisie. Il est important de souligner qu'une table de faits ne peut pas contenir des faits dont le grain serait différent.

Ralph KIMBALL recommande dans [Kimball and Ross, 2013] de choisir le grain le plus fin possible dans le processus métier. Ce qu'il nomme un **grain atomique** et qui va permettre d'être suffisamment robuste pour répondre aux requêtes imprévisibles des utilisateurs.

1.3.3 Les dimensions

Les dimensions sont des **axes d'observation** avec lesquels l'analyse des faits va être réalisée. Ces axes permettent de *classifier* les faits en fonction du temps, de la localisation ou bien d'autres critères. C'est le croisement des dimensions qui va permettre d'analyser les faits selon plusieurs perspectives. Les dimensions vont fournir le contexte des faits (qui, quoi, où, quand, pourquoi et comment).

Par analogie, une dimension est similaire à un arbre, l'objectif étant de joindre les faits à une feuille ou un nœud de cet arbre. En créant ce lien entre mesures et dimensions, les analystes seront capables de calculer des agrégats par nœud, par branche ou pour l'ensemble du tronc, ceci sur plusieurs dimensions.

Une dimension dans un entrepôt de données est composée de **membres**

regroupés en fonction des objets clés de l'entreprise. Un membre est caractérisé par **un ensemble d'attributs** et il est représenté comme une instance dans cette dimension, ce qui se traduira dans l'implémentation par une ligne dans la table de dimension. Dans la dimension géographique par exemple nous retrouverons toutes les villes mais également les régions et les pays. *Namur/Wallonie/Belgique* ou bien *Paris/Ile-de-France/France* sont des membres de la dimension.

Comme on peut le voir sur la figure 1.3, ces attributs ont pour rôle de **filtrer les requêtes** (ex : ville, année,...), et d'étiqueter les résultats. Plus les membres de dimensions seront garnis d'attributs riches et de qualité, plus l'entrepôt de données sera puissant pour les analyses décisionnelles.

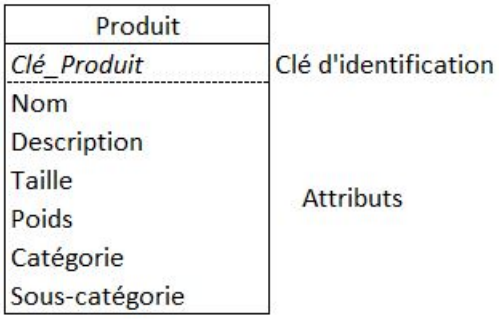


FIGURE 1.3 – Exemple d'une table de dimension produit

Parmi les différences entre une mesure d'une table de fait et un attribut d'une table de dimension, nous pouvons dire qu'une mesure est dépendante d'un événement enregistré dans l'entreprise alors qu'un attribut s'affranchit de ces événements. Une autre différence est que les mesures servent dans le calcul d'indicateurs de performance alors que les attributs de dimensions servent à filtrer les faits ou à les étiqueter.

Hiérarchies dimensionnelles

Quand un ensemble d'attributs a une relation hiérarchique dans une dimension, nous sommes en présence d'une **hiérarchie dimensionnelle**. Les hiérarchies définissent des chemins d'accès dans les données et se présentent sous une forme simple ou multiple.

Un exemple de hiérarchie dimensionnelle simple serait *pays* ⇒ *régions* ⇒ *provinces* ⇒ *villes*. On retrouve donc ici une hiérarchie naturelle dans une dimension géographique avec des attributs pays, région, province et ville. Les hié-

rarchies dimensionnelles multiples sont présentent quand, dans une dimension, il est possible de parcourir les données suivant plusieurs "*chemins*". En prenant pour exemple la dimension temporelle, celle-ci peut contenir une hiérarchie *année* ⇒ *trimestre* ⇒ *mois* ⇒ *jour* mais également une hiérarchie basée sur l'année fiscale et qui serait *année fiscale* ⇒ *trimestre fiscal* ⇒ *mois fiscal* ⇒ *jour*. Dans tous les cas, ces hiérarchies dimensionnelles multiples peuvent coexister dans une seule dimension.

1.3.4 Le schéma en étoile

Un schéma en étoile [Figure 1.4] est, dans sa forme la plus simple, constitué d'une table regroupant les faits et d'une ou plusieurs tables de dimensions liées à la table de faits. Dans un modèle dimensionnel, seul la table des faits possède des relations avec les tables de dimensions.

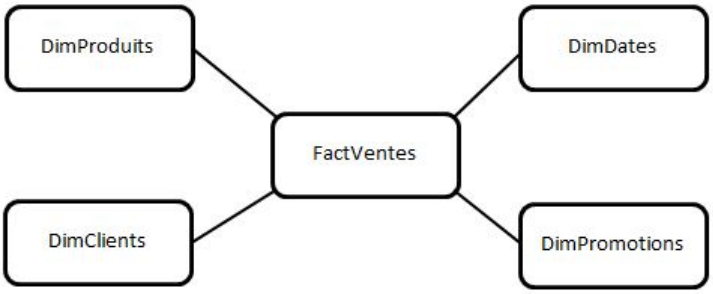


FIGURE 1.4 – Exemple de schéma en étoile

Le modèle dimensionnel résultant de ce regroupement de dimensions autour d'une table de faits va offrir à l'utilisateur une meilleure compréhension du problème. En effet, l'utilisateur va pouvoir très rapidement identifier les filtres à choisir afin d'accéder à l'information qu'il recherche.

Un autre avantage du modèle dimensionnel en étoile est de réduire le nombre de jointures lors de l'implémentation dans une base de données. Ce n'est pas négligeable lorsque l'on souhaite parcourir une table de faits contenant un très grand nombre de données.

1.3.5 Le schéma en flocon

Un schéma en flocon [Figure 1.5] n'est pas fondamentalement différent d'un schéma en étoile. La principale différence se situe au niveau des relations

pouvant exister entre deux dimensions. Le schéma en flocon se compose d'une table de faits connectée à plusieurs tables de dimensions qui peuvent elles-mêmes être connectées à d'autres tables de dimensions via une relation n à un.

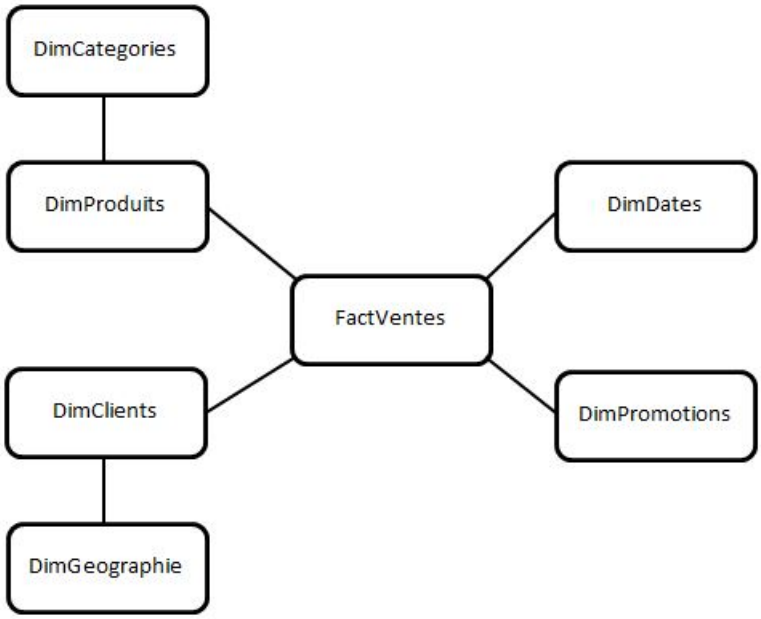


FIGURE 1.5 – Exemple de schéma en flocon

Ce type de design est utilisé lorsqu'on souhaite normaliser certaines tables de dimensions. Un des avantages de ce type de schéma est qu'il va permettre de mettre à jour des dimensions en cas de changement sans devoir modifier tous les champs d'un attribut dans la table de la dimension. Un autre avantage est le gain d'espace du fait de la non-redondance de certaines informations suite à la normalisation de la dimension. Cependant cette économie d'espace est relative puisque l'entrepôt de données est majoritairement composé des informations contenues dans les tables de fait plutôt que dans les tables de dimensions. En termes d'inconvénients, le schéma en flocon peut se révéler moins intuitif pour l'utilisateur et il peut également mener à une dégradation des performances sur le temps de réponse dûe au nombre plus important de jointures additionnelles.

1.4 La modélisation Hainaut d'un problème de calcul

Dans son ouvrage [Hainaut, 2002], nous pouvons trouver une méthode de conception d'un modèle sous une forme d'expression abstraite et pouvant ensuite être traduit dans un tableur de type Excel.

Pour réaliser ce modèle, la méthode de Jean-Luc HAINAUT consiste en une démarche en plusieurs étapes et utilise des concepts de base qui sont réduits à leur plus simple expression. Le modèle est composé de **grandeurs** qui pour certaines sont de type **données** et d'autres de type **résultats**. Le modèle peut comporter des **grandeurs internes** servant de données intermédiaires dont l'existence n'est pas perçue par l'utilisateur. Le modèle est également composé de **règles** permettant d'exprimer un résultat en fonction des grandeurs. Ce sont ces règles qui font office de **relations** entre les grandeurs.

Une grandeur dans le modèle possède une **valeur** et est le plus souvent mesurable. La valeur d'une grandeur peut être de différents types : le plus souvent ces valeurs seront de type **numérique** (quantité commandée, nombre d'employés,...) mais également de type **logique** (vrai/faux) ou bien **qualitative** et qui correspondent à un libellé tel qu'un nom de ville, une catégorie de produit, etc.

Le modèle est dit simple quand les grandeurs ne peuvent prendre qu'une seule valeur à la fois. À l'opposé, le modèle est dit dimensionné lorsqu'une grandeur possède simultanément plusieurs valeurs en fonction d'une ou plusieurs **dimension(s)**.

La démarche décrite dans [Hainaut, 2002] suit des principes qui vont guider l'analyse du problème. Ces principes consistent d'une part à penser en termes de règles très simples à élaborer, quitte à définir des grandeurs internes, et d'autre part à partir des résultats pour remonter vers les données. Cette démarche sera développée dans le chapitre 3.

Dans le cadre de notre approche visant à transformer un problème de calcul à plusieurs dimensions en une solution dans un tableur domestique, le modèle résultant de la démarche de Jean-Luc HAINAUT sera utile pour définir et implémenter les règles de calcul dans des formules DAX afin d'obtenir des mesures calculées ou des colonnes calculées sur la base des données présentes dans l'entrepôt de données.

1.5 En résumé

A travers ce chapitre nous avons pu voir deux types de modélisations qui auront tous les deux leur utilité lors de la transformation d'un problème en une solution dans un tableur. Le **modèle dimensionnel** va servir à identifier les faits et les dimensions et organiser notre entrepôt de données tandis que le modèle de calcul de Jean-Luc HAINAUT, que je vais nommer **modèle Hainaut**, va établir les règles de calculs et d'agrégations qui définiront les mesures calculées à ajouter à notre modèle dimensionnel.

Durant ce chapitre, nous avons pu aborder des concepts propres aux deux types de modélisation et pourtant des liens entre ces concepts sont réalisables. Dans les deux modélisations, un terme commun est celui de **dimension**. Une dimension représente en effet le même concept que ce soit dans le modèle dimensionnel ou bien dans le modèle Hainaut. En ce qui concerne les concepts de **mesures**, d'**attributs** et de **grandeurs**, la différence est plus discrète. Dans le modèle Hainaut, tout est question de grandeurs et ce qu'il soit question de données en entrée, des grandeurs internes ou bien des résultats. Pourtant, si on compare les deux modèles, on peut remarquer que les **résultats** du modèle Hainaut correspondent à des **mesures calculées** dans nos tables de faits du modèle dimensionnel. Pour ce qui est des **grandeurs internes**, celles-ci sont le résultat d'une règle interne non perçue par l'utilisateur et elles peuvent être rattachées également à des **mesures calculées** ou bien des **colonnes calculées**. Enfin, pour ce qui est des **attributs** d'une dimension dans le modèle dimensionnel, ceux-ci vont correspondre à des **grandeurs** de type *qualitatif* et qui représenteront le plus souvent un libellé.

Nous verrons plus tard dans ce mémoire comment il va être possible de concevoir ces deux modèles et de les combiner afin d'obtenir notre entrepôt de données et la possibilité de naviguer à travers celui-ci dans un tableur Excel.

Chapitre 2

Conception d'un modèle dimensionnel

2.1 Introduction

Lors du chapitre précédent nous avons abordé la théorie concernant la modélisation dimensionnelle. Pour rappel, il n'existe pas un seul et unique modèle dimensionnel qui a les faveurs unanimes de la communauté scientifique. J'ai donc choisi la méthode qui est décrite dans [Kimball and Ross, 2013] afin de réaliser un modèle dimensionnel. Si nous devions faire une comparaison avec la réalisation de modèles pour un système de base de données relationnelles, le modèle dimensionnel résultant de cette méthode est un modèle à mi-chemin entre un modèle conceptuel et un modèle logique.

L'avantage de ce modèle est qu'il est simple à comprendre pour un utilisateur n'ayant pas de connaissances en modélisation de base de données multidimensionnelles et qu'il ne nécessite pas l'apprentissage d'une notation spécifique telle que la notation MultiDim décrite dans [Malinowski and Zimányi, 2008]. Ce modèle dimensionnel prend place dans le cycle de vie d'un projet décisionnel entre les exigences utilisateurs et le modèle physique de données comme on peut le voir sur la figure 2.1.

Dans [Kimball and Ross, 2013] une méthode en étapes nous est fournie pour la réalisation de ce modèle. Cette méthode va consister à définir les **processus métiers** à analyser dans la problématique donnée, puis de définir la **granularité** des faits qui seront analysés. Ensuite il s'agira d'identifier les **dimensions** qui se rapporteront aux faits, de définir les attributs de ces dimensions ainsi que les hiérarchies et, finalement, d'identifier les **faits** qui seront enregistrés

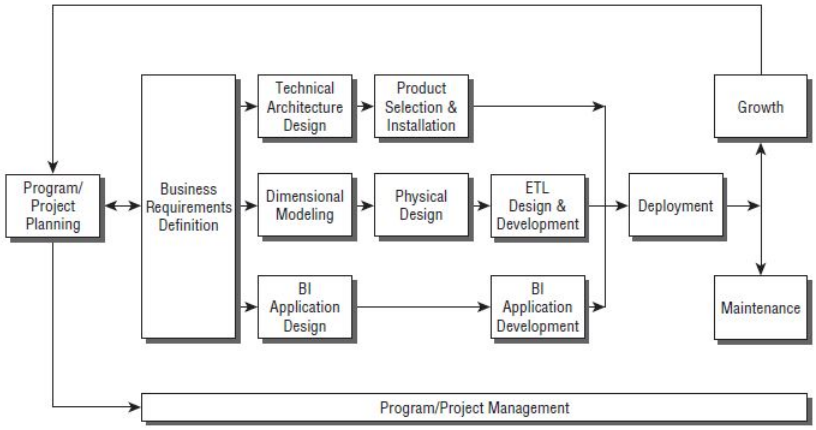


FIGURE 2.1 – Le cycle de vie d'un projet BI repris depuis l'ouvrage [Kimball and Ross, 2013]

dans l'entrepôt de données. De cette démarche résultera un schéma en étoile permettant de servir de base pour la réalisation du modèle Hainaut.

2.2 Sélection du processus métier

Une des premières étapes dans la construction d'un modèle dimensionnel est de définir le ou les processus métiers qui devront être analysés. Un processus métier rassemble les activités opérationnelles de l'entreprise qui vont générer des informations. Celles-ci deviendront des faits dans le système.

Nous avons pu voir dans la section 1.3 qu'un modèle dimensionnel comprend autant de tables de faits que de processus métier à analyser. Dans le cadre de grands entrepôts de données de sociétés, ceux-ci peuvent comporter un très grand nombre de processus métier. Citons en exemple une entreprise générant des informations aussi bien sur les ventes qu'elle réalise que sur la production des biens qu'elle fabrique. Nous avons donc deux processus métier qui donneront dans notre modèle deux tables de faits et des dimensions. Certaines de ces dimensions seront communes aux tables de faits telle que la dimension produit qui pourra être aussi bien liée aux ventes mais également à la production.

Quelques conseils sont donnés dans [Kimball and Ross, 2013] afin d'identifier les processus métier :

- un processus métier est le plus souvent une **activité** réalisée par l'entreprise. Il est donc important d'identifier à la lecture de la problématique les *verbes* qui sont utilisés ;
- un processus métier implique typiquement un **système opérationnel** dans l'entreprise tel qu'un système de facturation ou de gestion des stocks ;
- un processus métier va **générer** ou bien **capturer** des mesures de l'activité de l'entreprise ;
- dans une entreprise il est probable qu'un processus métier soit la suite d'un autre processus métier. Les données résultant d'un premier processus deviendront les données entrant dans un autre. Cette chaine de processus va donc nous conduire à la réalisation de plusieurs tables de faits.

En prenant en exemple la société de vente de vélos et d'accessoires que nous avions déjà évoquée dans le chapitre 1, on peut identifier parmi les processus métiers celui des ventes de produits fabriqués par l'entreprise auprès des revendeurs.

Choisir les processus métiers découle donc de l'analyse de la problématique à résoudre. À ce titre, il est important de communiquer avec les utilisateurs afin de ne pas considérer la vision stratégique de l'entreprise comme un des processus métier mais bien d'analyser les actions concrètes au sein de l'entreprise. Dans le cadre de ce mémoire, nous avons choisi une approche domestique, ce qui veut dire que nous sommes en présence de problèmes de complexité moyenne et dont les processus métiers sont supposés être plus restreints par rapport à ceux d'une moyenne ou grande entreprise. Les processus métiers devraient donc être facilement identifiables à la lecture de l'énoncé de la problématique.

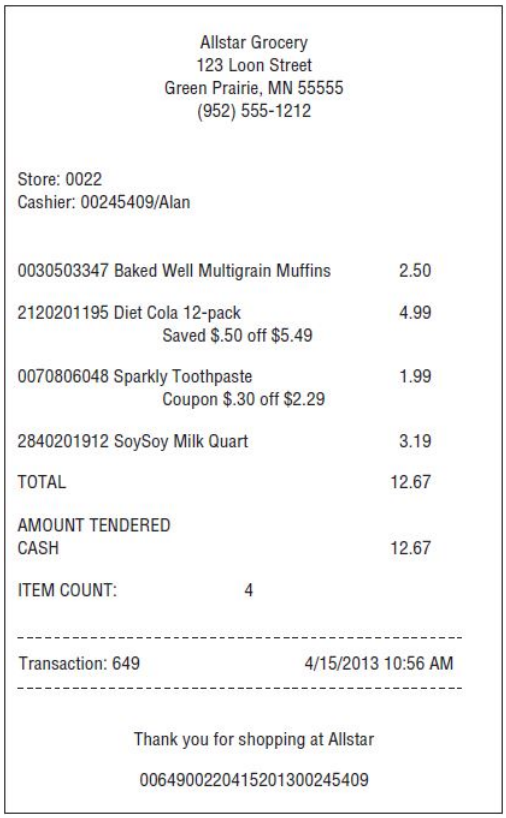
2.3 Définir la granularité

Comme nous avons pu le voir dans la section 1.3.2, le grain correspond au niveau de détail des mesures d'un fait dans une table de faits. La granularité d'un fait est propre à chaque table de faits. Il n'est en effet pas conseillé d'avoir dans une même table de faits des grains différents au risque de générer des erreurs dans certaines mesures agrégées.

Choisir la granularité pourrait se résumer à simplement choisir le grain le plus fin (**grain atomique**) du fait découlant d'un événement de notre processus métier. Choisir le grain atomique va permettre de capter l'ensemble des informations générées par l'événement survenant dans un processus métier et il

sera bien plus robuste lorsqu'un utilisateur souhaitera effectuer des *drill-down* dans l'entrepôt de données.

Il est possible de définir une granularité moins détaillée et dont les mesures dans la table de faits seraient dès lors des agrégats de différentes mesures générées par un système opérationnel. Le risque serait qu'un utilisateur se heurte à un mur lorsqu'il souhaitera naviguer plus en détail dans les faits.



Allstar Grocery
123 Loon Street
Green Prairie, MN 55555
(952) 555-1212

Store: 0022
Cashier: 00245409/Alan

0030503347 Baked Well Multigrain Muffins 2.50

2120201195 Diet Cola 12-pack 4.99
Saved $.50 off $5.49

0070806048 Sparkly Toothpaste 1.99
Coupon $.30 off $2.29

2840201912 SoySoy Milk Quart 3.19

TOTAL 12.67

AMOUNT TENDERED
CASH 12.67

ITEM COUNT: 4

Transaction: 649 4/15/2013 10:56 AM

Thank you for shopping at Allstar

00649002204152013 00245409

FIGURE 2.2 – Exemple de ticket de caisse repris depuis l'ouvrage [Kimball and Ross, 2013]

En prenant en exemple un ticket de caisse, il est possible d'avoir un ticket qui indiquerait, pour chaque produit acheté, sa quantité, son montant et, à la fin du ticket, le montant total des articles comme sur la figure 2.2.
Il est également possible d'avoir un autre type de ticket de caisse qui lui ne contiendrait que le montant total des articles achetés. Dans les deux cas, la vente représentera un fait dans notre entrepôt de données mais, en fonction des informations contenues sur le ticket, il sera possible d'affiner nos analyses décisionnelles. C'est ici que la granularité aura son importance. Si les ventes

enregistrées ne reprennent pas le détail des articles achetés, il ne sera pas possible d'établir l'analyse des ventes par produits achetés.

Exprimer le grain d'un modèle dimensionnel ne consiste pas à simplement répertorier la liste des dimensions qui seront reliées au fait mais bien à exprimer dans des termes métiers ce qu'il représente.

Le grain doit être choisi dès que l'analyse des processus métiers sera terminée. Il est important de ne pas passer outre cette étape lors de la conception du modèle. Une granularité mal définie risque à terme d'impliquer un retour en arrière dans la conception du modèle lors des étapes de définition des dimensions et des faits. Ceci peut bien entendu provoquer des retards et des risques d'erreurs de conceptions à la clé.

2.4 Identification des dimensions

Les dimensions sont des axes d'analyses qui vont avoir pour but de fournir une manière de filtrer les faits dans une table de faits. Chaque table de faits va être reliée par un ensemble de dimensions.

Identifier les dimensions peut être facilement réalisable dès que la granularité a été correctement définie. Ces dimensions doivent être le résultat des questions de type "qui", "que", "quoi", "quand", "où", "comment" et "pourquoi" concernant les mesures contenues dans un fait. Dans notre exemple d'entreprise de vente de vélos, un fait de la table de faits "vente" correspond à une vente réalisée par un client, pour un produit, à un moment donné. Nous sommes donc ici en présence de trois dimensions qui vont caractériser le fait et qui sont les dimensions "Client", "Produit" et "Date".

Si l'on souhaite ajouter d'autres dimensions à notre table de faits, il est important de veiller à ce que la granularité soit respectée. Ceci veut dire que le choix d'un membre de cette dimension à ajouter, cumulé avec les choix de membres des autres dimensions, doit toujours correspondre à un et un seul fait dans la table des faits. Si tel n'était pas le cas, il est fort probable que cette dimension additionnelle ne corresponde pas à la granularité qui aura été définie précédemment. Elle devra être retirée ou bien le grain devra être redéfini.

2.4.1 Les attributs de dimension

Une fois les dimensions identifiées, il est nécessaire de lister l'ensemble des **attributs** qui sont contenus dans ces dimensions. Ces attributs peuvent être

de type *descriptif* ou bien de type *numérique* et contiendront des valeurs qui seront utilisées pour filtrer les faits lorsque l'utilisateur désirera naviguer dans l'entrepôt de données. Dans notre exemple précédent, la dimension "Produit" va pouvoir contenir plusieurs attributs en rapport aux produits mis en vente par l'entreprise.

Certains attributs vont être de type descriptif et contenir du texte tel que le nom du produit ou bien la marque du produit. Ces attributs vont pouvoir servir à présenter mais également à filtrer les faits. On pourrait dès lors demander d'afficher le montant total des ventes réalisées en fonction des marques vendues par l'entreprise.

Un attribut peut également être de type numérique et va représenter une valeur numérique servant à filtrer ou grouper des faits. Si l'on prend en exemple des chaussures vendues par l'entreprise, ces chaussures sont caractérisées par une pointure qui est une valeur numérique. Cette pointure va nous permettre entre autre de consulter le nombre de ventes réalisées sur des chaussures en filtrant le nombre de ventes par pointure. Il n'y a pas de calculs à proprement dit sur ces pointures, la pointure sera donc considérée comme un attribut de la dimension produit et non pas une mesure dans la table de faits vente.

Dans le cas où un attribut serait spécifique à un sous-ensemble des membres de la dimension, il n'est pas nécessaire de vouloir soustraire ces membres pour créer une dimension spécifique. La solution est d'ajouter une valeur qui serait commune pour tous les membres non concernés par cet attribut. En reprenant l'exemple des produits vendus par une entreprise, l'attribut "pointure" est spécifique aux produits de la catégorie chaussures. Pour tous les autres produits qui ne seraient pas des chaussures, il suffira d'encoder une valeur tel que "N/A" dans le champs pointure.

2.4.2 Hiérarchies dimensionnelles

Parmi l'ensemble des attributs définis dans une dimension, certains seront liés entre eux. Ces attributs vont entrer dans la composition d'une **hiérarchie dimensionnelle**. Dans une hiérarchie dimensionnelle, nous retrouvons donc des attributs pour lesquels un lien de subordination est présent. Dans la dimension temporelle "Date", les attributs "jour", "mois", "trimestre" et "année" sont reliés entre eux. Une année comporte quatre trimestres qui chacun comporte trois mois et qui eux contiennent entre vingt-huit et trente et un jours.

Dans une table de dimension d'un entrepôt de données, les attributs qui composent une hiérarchie sont dans une seule et unique table. Il n'y a pas de

normalisation comme dans un modèle entité-relation. Ce choix de ne pas normaliser les dimensions comme le recommande [Kimball and Ross, 2013] permet de répondre à deux objectifs de la modélisation dimensionnelle qui sont la rapidité et la simplicité. Chaque ligne de la table de dimension va donc contenir un membre dont les attributs de la hiérarchie seront organisés d'une manière aplatie tel qu'on peut le voir dans la figure 2.3.

	CléGéographie	Ville	Province	Pays	CodePostal	ZoneGéographique
217	216	Courbevoie	Hauts de Seine	France	92400	Europe
218	217	Paris La Defense	Hauts de Seine	France	92081	Europe
219	218	Sèvres	Hauts de Seine	France	92310	Europe
220	219	Suresnes	Hauts de Seine	France	92150	Europe
221	220	Bobigny	Seine Saint Denis	France	93000	Europe
222	221	Drancy	Seine Saint Denis	France	93700	Europe
223	222	Pantin	Seine Saint Denis	France	93500	Europe
224	223	Saint-Denis	Seine Saint Denis	France	93400	Europe
225	224	Tremblay-en-France	Seine Saint Denis	France	93290	Europe
226	225	Orly	Val de Marne	France	94310	Europe
227	226	Cergy	Val d'Oise	France	95000	Europe
228	227	Abingdon	England	Royaume-Uni	OX14 4SE	Europe
229	228	Basingstoke Hants	England	Royaume-Uni	RG24 8PL	Europe
230	229	Berks	England	Royaume-Uni	SL4 1RH	Europe
231	230	Berkshire	England	Royaume-Uni	RG11 5TP	Europe

FIGURE 2.3 – Exemple d'une dimension géographique

D'après [Da Costa, 2011], il n'est pas conseillé d'avoir des relations n vers n dans une dimension pour la création de hiérarchies. Il est préférable de privilégier les relations 1 à n afin de faciliter la création de rapports et d'agrégations. Par exemple : une ville est située dans une région qui est elle même située dans un pays. Comme il est indiqué dans [Da Costa, 2011], la création des hiérarchies dans une dimension est similaire à des poupées russes qui viendraient s'emboiter les unes dans les autres.

2.4.3 Conseils lors de la conception des dimensions du modèle

Les conseils suivants sont en partie issus de [Kimball and Ross, 2013] et de [Da Costa, 2011] .

Une dimension peut évoluer et certains membres peuvent être amenés à ne plus être utilisés (un produit retiré de la vente par exemple ou bien l'attribut "catégorie" d'un membre produit pourrait être modifié). Tout comme de nouveaux membres peuvent être ajoutés durant le cycle de vie de l'entrepôt de données. Ce type de dimensions est aussi nommé *dimensions à évolution lente* et peut être traité de plusieurs manières d'après [Kimball and Ross, 2013]. Carlos DA COSTA conseille de ne pas supprimer les membres dans une dimension

afin de ne pas rompre l'historique des faits reliés aux membres supprimés, tout comme il vaut mieux ne pas mettre à jour un attribut en écrasant l'ancienne valeur par la nouvelle valeur. En revanche il conseille lors de la conception de la dimension, d'ajouter un attribut "état" indiquant le statut du membre dans la dimension. Un autre conseil est d'ajouter un attribut de type *date* qui sera mis à jour avec la date où le membre ne sera plus considéré comme actif dans le système.

Un membre dans une dimension est unique dans la dimension. Si on observe que deux membres dont les valeurs dans les attributs sont identiques, il est fort probable que nous soyons en présence d'un même membre et il faudra supprimer les doublons.

Il est conseillé de n'établir des relations entre les dimensions que si c'est nécessaire. L'avantage d'une modélisation dimensionnelle telle que décrite par [Kimball and Ross, 2013] est que la normalisation des dimensions n'est pas une obligation contrairement à un modèle relationnel. Ceci permet au modèle de rester simple et compréhensible pour les utilisateurs.

2.5 Identification des faits

La dernière étape dans la conception du modèle dimensionnel va être d'identifier les faits devant apparaître dans les tables de faits. Tout comme pour l'identification des dimensions, l'identification des faits va être guidée par la granularité que nous avons définie lors de la deuxième étape. Un fait doit toujours être consistant avec le grain. C'est en répondant à la question "Qu'est ce que le processus métier mesure?" que l'on va pouvoir déterminer les faits. Un fait typique va contenir des mesures telle qu'une quantité commandée ou bien le coût de production d'un produit.

L'identification des faits va en grande partie être guidée par l'analyse des données générées par les systèmes opérationnels de l'entreprise.
Il est important de tenir également en compte les demandes des utilisateurs qui seront amenés à utiliser l'entrepôt de données, ces demandes seront représentées en tant que mesures calculées dans l'entrepôt de données. Cette étape d'identification des mesures calculées sera réalisée plus tard dans notre approche sur base des grandeurs intermédiaires et des résultats qui seront produits par le modèle Hainaut.

Les mesures contenues dans une table de faits sont classées en trois catégories : les mesures *additives*, *semi-additives* et *non-additives*. Les mesures **additives** peuvent être agrégées selon n'importe quelle dimension dont dépend le

fait (les quantités d'articles vendus ou le montant total des ventes par exemple). Les mesures **semi-additives** ne peuvent être agrégées que sur certaines dimensions (un solde de compte agrégeable par clients mais pas sur la dimension temporelle). Enfin les mesures **non-additives** sont celles qui ne peuvent pas être agrégées (le coût unitaire d'un produit). On peut retrouver dans la notation MultiDim [Malinowski and Zimányi, 2008] une notation permettant d'identifier le type de mesure lors de la conception du modèle tandis que rien n'est spécifié dans le modèle de [Kimball and Ross, 2013].

Il arrive parfois qu'un fait ne soit pas concerné par une dimension. On pourrait dès lors être tenté de mettre la valeur **NULL** dans le champs correspondant à la clé étrangère pointant vers cette dimension. Cette façon de faire n'est pas conseillée car le risque serait de générer des erreurs de calcul lors d'agrégations de mesures de faits. Il est plutôt préférable d'ajouter, dans la dimension non concernée par le fait, un membre spécial qui représenterait une valeur *non applicable*. Par exemple, il est possible qu'une vente ne soit pas concernée par une promotion et donc que le fait ne soit pas relié à la dimension "Promotion". En ajoutant dans la dimension un membre "Pas de promotion" et en garnissant le reste des attributs de cette dimension, on pourra relier la vente n'accordant pas de promotion à la dimension.

En suivant les recommandations que l'on peut retrouver dans [Da Costa, 2011], voici quelques conseils pour la réalisation de tables de faits :

- veillez à l'unicité des faits dans la table et pour ce faire, choisissez une clé résultant de la concaténation des clés des dimensions car si un fait a les mêmes valeurs de dimensions qu'un autre fait, alors ce doit être le même enregistrement dans la table ;
- la dimension temps est quasiment utilisée dans chaque table de faits. Puisqu'un fait représente un événement dans la vie de l'entreprise, ce fait survient toujours à un moment donné. Il y a donc une dimension temporelle qui le caractérise. Cette dimension temporelle par rapport aux faits permet d'obtenir un historique des mesures ;
- parce que les données contenues dans une table de faits représentent une situation passée, ces données sont figées. Il n'est donc pas permis de faire de mises à jour des données afin que les données restent cohérentes et similaires peu importe le moment où l'analyse à lieu. Si un analyste réalise un rapport à un moment donné et qu'il souhaite générer le même rapport un mois (ou un an) après, les résultats dans le rapport doivent être identiques ;
- une table de faits contient (presque) toujours des mesures de types nu-

mériques et additives. Évitez de stocker des moyennes car une somme de moyennes n'est pas équivalente à la moyenne des sommes et ceci est valable également pour des ratios et pourcentages ;

- si le croisement des dimensions reliant la table de faits ne produit aucun fait, alors il n'est pas nécessaire de stocker un fait dont les mesures sont égales à zéro. Stocker des données comme étant le produit cartésien de dimensions n'a pas d'utilité mais en plus viendra alourdir fortement la table de faits et rendra sa maintenance difficile. Par exemple, si l'entreprise n'a pas vendu de produits durant une journée, il n'y a pas lieu d'avoir dans la table de faits des enregistrements indiquant pour tous les produits et pour tous les revendeurs une quantité vendue égale à zéro. Pour rappel, une table de faits ne contient que des enregistrements d'événements survenus dans la vie de l'entreprise.

2.6 En résumé

Dans ce chapitre nous avons pu voir comment réaliser un modèle dimensionnel en suivant les étapes de réalisation recommandées dans [Kimball and Ross, 2013]. Un exemple de résultat de la création d'un modèle dimensionnel grâce à cette méthodologie ressemblera à la figure 2.4 dans le cas de la modélisation des ventes d'une entreprise de distribution. La notation de ce modèle va dans le cadre du mémoire être légèrement modifiée par la suite dans le chapitre 5 quand il sera question de créer un modèle par un public n'ayant que peu d'expérience dans la modélisation de problèmes informatiques.

Le modèle obtenu en suivant les étapes décrites dans ce chapitre ne prend pas en compte des cas particuliers de tables de faits ou de dimensions spécifiques telles que des dimensions dégénérées, des dimensions à évolution lente ou bien des tables de pont dans le cas de dimensions multi-valuées. Il est néanmoins possible de trouver dans [Kimball and Ross, 2013] ou bien dans [Malinowski and Zimányi, 2008] les moyens à mettre en œuvre pour réaliser ces modèles spécifiques.

Il est possible de retrouver dans l'ouvrage [Malinowski and Zimányi, 2008] une autre méthode de réalisation pour la création d'un entrepôt de données qui va passer par des étapes de conceptions de modèles conceptuels, logiques et physiques à la manière des systèmes relationnels de base de données. Cette méthode requiert pourtant à mon sens des compétences en modélisation qui dépassent celles du public visé par ce mémoire.

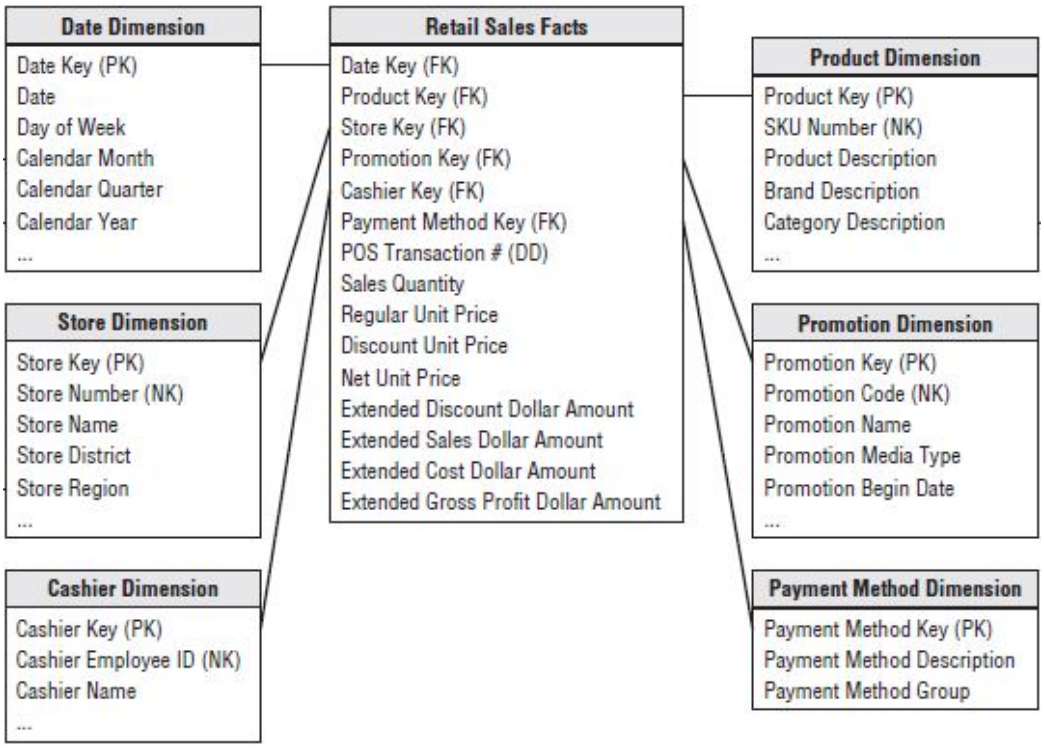


FIGURE 2.4 – Exemple d'un modèle dimensionnel tel que repris dans [Kimball and Ross, 2013]

La conception du modèle présentée dans ce chapitre ne reprend pas les mesures calculées ni la manière de les traiter, mais se limite à l'organisation des données. Les mesures ainsi que les calculs pour obtenir les résultats seront présentés dans le chapitre suivant grâce au modèle Hainaut.

Chapitre 3

Conception d'un modèle Hainaut

3.1 Introduction

En suivant les recommandations et les précèptes donnés dans [Hainaut, 2002], nous allons réaliser une modélisation de l'énoncé en se concentrant tout particulièrement sur les résultats attendus et les règles permettant d'y arriver. La conception du modèle Hainaut présenté ici part du principe que l'ensemble des grandeurs et des dimensions qui seront utilisées pour l'obtention des résultats a déjà été défini lors de la conception du modèle dimensionnel.

Dans son ouvrage, Jean-Luc HAINAUT propose une démarche en trois phases pour réaliser la modélisation d'un problème de calcul : l'**analyse**, la **normalisation** et la **validation**.

3.2 Analyse et conception du modèle

Durant la phase d'analyse nous allons construire un modèle en partant des principes énoncés par Jean-Luc HAINAUT.
Le *premier principe* consiste à construire le modèle en partant des résultats et en remontant depuis ceux-ci vers les données.
Le *deuxième principe* recommande de ne formuler que des règles simples quitte à devoir définir des grandeurs internes intermédiaires.
Le *troisième principe* est de lister l'ensemble des données en entrée afin de vérifier que les règles formulées par le premier principe utilisent bien l'ensemble des données listées. Si tel ne devait pas être le cas, ces données peuvent être considérées comme facultatives et retirées du modèle.

Le *quatrième principe* indique la fin du processus d'analyse. Celui-ci se termine lorsque, parmi les règles déjà élaborées, toutes les grandeurs qui se trouvent en partie droite d'une règle sont soit une donnée ou bien une grandeur qui apparaît en partie gauche d'une autre règle.

De ces quatre principes, le troisième ne sera pas respecté dans notre approche à ce stade de la conception puisque l'ensemble des données en entrée a déjà été défini lors de la conception du modèle dimensionnel. L'avantage est un gain de temps dans la conception du modèle Hainaut. Ce principe sera mis en application par la suite lorsqu'il sera question de réviser le schéma en étoile du modèle dimensionnel.

Exemple Notre entreprise va vendre des produits à des clients et en fonction de la quantité d'un article commandé, le client va se voir attribuer une remise (en pourcentage) sur sa commande. L'entreprise souhaite connaitre le montant à facturer pour une quantité d'un article à un client. Il n'y a donc pas de dimensions à prendre en compte dans notre exemple. Le modèle Hainaut correspondant à l'énoncé de ce problème sera :

Résultats
MONTANT : réel (€) ; Montant à payer par le client
Grandeurs internes
SOUS_MONTANT : réel (€) ; Montant des articles à payer sans prendre en compte la remise
MONTANT_REMISE : réel (€) ; Montant de la remise à déduire
Règles
MONTANT = SOUS_MONTANT - MONTANT_REMISE
SOUS_MONTANT = PRIX * QUANTITÉ
MONTANT_REMISE = SOUS_MONTANT * (1-REMISE)

Dans cet exemple nous avons pu voir que les règles ont été formulées afin d'être le plus simples possibles et pour cela nous avons fait appel à deux grandeurs internes qui sont SOUS_MONTANT et MONTANT_REMISE. Nous avons également utilisé comme données en entrée des mesures factuelles du modèle dimensionnel conceptuel réalisé durant le chapitre précédant. Enfin, la conception des règles a été élaborée à partir du résultat qui était demandé tout en remontant vers les données en entrée. Nous avons donc dans cet exemple mis en pratique les quatre principes énoncés dans [Hainaut, 2002] afin de réaliser un modèle de notre problème.

3.2.1 Généralisation par dimensionnement

Dans le cas d'un problème de calcul contenant des dimensions, on peut trouver dans [Hainaut, 2002] une approche de généralisation par dimensionnement afin de simplifier l'analyse du problème. L'idée de cette généralisation par dimensionnement consiste à construire un modèle réduit décrivant une vue plus simple du problème en ignorant une ou plusieurs dimensions. Après avoir créé et validé ce modèle réduit, on le modifie afin d'y ajouter une des dimensions ignorées. Cette modification du modèle réduit se fait de la manière suivante :

- on identifie les grandeurs qui sont dépendantes de la dimension à ajouter et on dimensionne ces données ;
- on ajoute des grandeurs intermédiaires qui incluent des fonctions agrégatives sur base de la dimension à ajouter.
- on contrôle la cohérence des dimensions des règles afin de détecter des erreurs.

Exemple En reprenant l'exemple précédant, nous souhaitons connaitre le montant total des ventes qu'un client a commandé auprès de notre entreprise, quelque soit les produits achetés par ce client.

Nous pouvons voir ici que la notion de *produit* est apparue en tant que dimension au problème de calcul. Ce qui veut dire que le prix, la quantité et le pourcentage de remise va être différent en fonction des produits que le client a commandé. Les grandeurs en entrées vont donc voir leurs **valeurs** devenir une **suite de valeurs**. Ces grandeurs sont de type **grandeurs multivaluées** et sont représentées par une notation indicée par la dimension qui les concernent.

Résultats
MONTANT : réel (€) ; Montant à payer par le client
Grandeurs internes
SOUS_MONTANT$_P$: réel (€) ; Montant des articles à payer sans prendre en compte la remise
MONTANT_REMISE$_P$: réel (€) ; Montant de la remise à déduire
MONTANT$_P$: réel (€) ; Montant à payer pour un produit déduit du montant de la remise accordée à ce produit

Règles
MONTANT = $\sum_p$ MONTANT$_P$

MONTANT$_P$ = SOUS_MONTANT$_P$ - MONTANT_REMISE$_P$
SOUS_MONTANT$_P$ = PRIX$_P$ * QUANTITÉ$_P$
MONTANT_REMISE$_P$ = SOUS_MONTANT$_P$ * (1-REMISE$_P$)

Dans cet exemple, nous pouvons voir que nous sommes parti du modèle de l'exemple précédent et nous y avons ajouté la dimension *produit*. Le résultat n'étant plus le montant à payer pour un produit mais bien pour l'ensemble des produits commandés par le client. Nous avons modifié le modèle réduit résultant de l'exemple précédant afin d'ajouter une grandeur interne MONTANT$_P$ et une règle de calcul qui va faire la somme des MONTANT$_P$ afin d'obtenir notre résultat final.

3.2.2 Les fonctions agrégatives

Dans l'exemple précédant, nous avons pu constater que le résultat obtenu est la somme des montants pour chaque produit. Nous avons donc utilisé dans la règle la fonction $\sum$. Cette fonction est dite **agrégative** car la valeur retournée est *agrégée* sur l'ensemble des valeurs de la grandeur multivaluée MONTANT$_P$.

Dans l'ouvrage [Hainaut, 2002], Jean-Luc HAINAUT liste d'autres fonctions agrégatives. Pour toute grandeurs (ou expression) G de type numérique et pour toute dimensions T non vide (contenant au moins une valeur), on considère la liste suivante comme représentative des fonctions agrégatives :

$\sum_T(G_T)$	: retourne la somme des G_T
$\prod_T(G_T)$	: retourne le produit des G_T
$\min_T(G_T)$	: retourne la valeur minimale des G_T
$\max_T(G_T)$	: retourne la valeur maximale des G_T
$\text{moy}_T(G_T)$	: retourne la valeur moyenne des G_T

Dans le cas où la grandeur est de type booléenne, les fonctions agrégatives suivantes sont à utiliser :

$\text{et}_T(G_T)$	: retourne la conjonction des G_T
$\text{ou}_T(G_T)$	: retourne la disjonction des G_T

Finalement, pour toute grandeur T, la fonction suivante est utilisable :

$\text{nombre}_T(G_T)$	: retourne le nombre de valeurs de la dimension T

En utilisant une fonction agrégative dans une règle ou une condition, cette fonction a comme propriété remarquable d'*absorber* la dimension sur laquelle elle est définie. Ce qui veut dire que le résultat de cette fonction perd la dimension de la fonction agrégative. En exemple, dans la règle :

$$\text{MONTANT} = \sum_{\text{P}} \text{MONTANT}_{\text{P}}$$

la valeur de la grandeur MONTANT n'est plus dépendant de la dimension P (produit), elle est devenue monovaluée.

3.2.3 Grandeurs à définition multiple

Dans le modèle de calcul, des grandeurs peuvent se voir attribuer des règles différentes en fonction de conditions, on les nomme des **grandeurs à définition multiples**. Prenons en exemple l'attribution d'un pourcentage de remise accordé à une commande respectant les conditions suivantes :

Une remise de 20% si la quantité est supérieure à 25
Une remise de 10% si la quantité est comprise entre 15 et 25
Aucune remise n'est accordée si la quantité est inférieure à 15

La définition de REMISE sera écrite sous la forme suivante :

REMISE = 0.2	si QUANTITÉ > 25
0.1	si 15 ≤ QUANTITÉ ≤ 25
0	sinon

Une attention particulière doit être apportée lors de la définition de grandeurs à définitions multiples. Trois propriétés essentielles doivent être respectées :

- la *complétude* : dans tous les cas, une grandeur à définition multiple doit avoir une valeur. Définir une branche *sinon* garanti cette complétude ;
- la *non-ambiguïté* : une grandeur à définition multiple ne doit être définie qu'une seule et unique fois. Il faut donc veiller à éviter des conditions ambiguës qui donneraient plusieurs valeurs à une même grandeur ;
- l'*absence de branches mortes* : Il ne peut y avoir dans la définition de conditions qui seraient toujours fausses.

3.3 Normalisation du modèle

La normalisation du modèle n'a pas pour but de détecter et corriger les erreurs du modèle (ce qui sera fait lors de la validation du modèle). Le but est ici d'améliorer la communication et la compréhension du modèle. Cela est d'autant plus important lorsque la complexité du modèle augmente.

La normalisation du modèle va passer par trois étapes : la *complétude*, l'*élimination de la redondance* et la *restructuration*.

- La complétude va permettre de vérifier si notre modèle est bien complet et que chaque grandeur aie un type, une unité, voire un commentaire ;
- l'élimination de la redondance va permettre de faire en sorte qu'une grandeur soit représentée une seule fois, car chaque grandeur représente un concept pertinent de notre domaine d'application. Si deux grandeurs ont la même règle de définition, l'une d'entre elle peut être retirée ;
- la restructuration permet de simplifier un modèle peu lisible et complexe en isolant certaines parties qui pourraient dès lors faire partie d'un sous-modèle. Ceci afin de pouvoir mieux faire évoluer le modèle dans le futur et faciliter sa validation.

3.4 Validation du modèle

Le modèle, une fois créé et normalisé, va devoir être validé afin de ne laisser aucune erreur de logique. Sans entrer dans le détail de la validation d'un modèle de calcul, deux étapes doivent être respectées : la **vérification de la cohérence** du modèle et le **test du modèle**.

La vérification de la cohérence du modèle va permettre de déceler la présence d'erreurs qui auraient été faites lors de l'étape d'analyse et de conception du modèle. D'après [Hainaut, 2002], il est recommandé de vérifier les éléments suivants :

- la validité de la structure globale du modèle ;
- la structure des règles de définition multiple ;
- la structure des règles de récurrence ;
- la cohérence des unités ;
- la cohérence des dimensions ;
- la cohérence des domaines de valeurs.

Tester le modèle va passer par la réalisation d'un jeu de données représentatives pour vérifier que les résultats qui seront produits par le modèle sont exacts. Ce même jeu de données pourra être utilisé lors de l'évolution future du modèle.

3.5 En résumé

Dans ce chapitre, nous avons pu voir comment réaliser un modèle permettant de résoudre un problème de calcul dont des grandeurs peuvent être dimensionnées. L'approche choisie pour obtenir ce modèle a été simplifiée par rapport à la théorie exposée dans [Hainaut, 2002] afin de correspondre à l'objectif de ce mémoire. Il n'est plus question ici de définir toutes les grandeurs et les dimensions utilisées par le modèle Hainaut puisque celles-ci l'ont déjà été dans le chapitre précédent lors de la réalisation du modèle dimensionnel.

Le modèle Hainaut tel que présenté ici se limite à définir les règles ainsi que les grandeurs internes et les résultats attendus. Dans le chapitre 5, nous verrons comment inclure ces grandeurs internes et ces résultats dans le modèle dimensionnel afin de correspondre aux mesures calculées. Les règles qui ont été définies dans ce modèle serviront par la suite à écrire les formules dans le langage DAX lors de la création de mesures calculées ou bien de colonnes calculées.

Chapitre 4

Outils d'analyse de données multidimensionnelles

4.1 Introduction

Dans ce chapitre nous allons décrire les outils mis à disposition des utilisateurs afin de pouvoir implémenter des problèmes contenant des données multidimensionnées. Parmi l'ensemble des outils disponibles pour la BI, nombreux sont ceux conçus pour un usage professionnel et dont le nombre d'informations à traiter est important et complexe. Dans le cadre de ce mémoire il a été convenu de se concentrer uniquement sur des outils accessibles aux utilisateurs ne possédant pas de grandes compétences en informatique. Nous resterons donc sur des problématiques de types domestiques et réalisables dans un tableur tel qu'Excel.

Nous aborderons en premier lieu la manière dont fonctionnent les solutions professionnelles pour traiter d'informatique décisionnelle. Ensuite, nous verrons comment dans le tableur Excel de Microsoft® l'outil PowerPivot permet de réaliser une solution similaire aux solutions professionnelles. La troisième partie du chapitre traitera des tableaux croisés dynamiques qui permettent d'effectuer différentes opérations sur un cube de données multidimensionnelles.

4.2 Les solutions professionnelles

Il existe, pour des problématiques plus complexes ou lorsque les données à manipuler deviennent plus importantes, des logiciels spécialisés dans la concep-

tion et la manipulation de base de données multidimensionnelles. Ces données sont dès lors rassemblées dans des entrepôts de données et, grâce à des outils OLAP (Online Analytical and Transactional Processing), peuvent être manipulées suivants différentes dimensions, elles même pouvant être organisées en hiérarchies. D'après Edgar Frank CODD, le mot OLAP désigne un ensemble de technologies permettant la prise de décision stratégique rapide et fiable sur des données modélisées en multidimensionnel.

Dans des solutions professionnelles traitant de Business Intelligence, la conception d'une solution est plus complexe. L'utilisation d'un outil ETL, qui va importer et traiter des données dans des bases de données relationnelles, va être nécessaire afin de rassembler des données venant de différentes sources, dans des formats parfois différents. Un autre outil permettant de définir des cubes OLAP devra également être utilisé afin de définir des mesures calculées, des vues ou des rôles pour exposer les informations aux utilisateurs en fonction de leurs profils. Un employé travaillant au service financier n'aura pas nécessairement besoin des mêmes informations contenues dans l'entrepôt de données qu'un autre travaillant au département marketing. C'est la raison pour laquelle l'accès aux données stockées au sein de l'entreprise ne sera pas directement accessibles à l'utilisateur mais passera par des vues prédéfinies lors de la construction du cube. A propos de la présentation des données et des résultats il sera possible dans des solutions professionnelles de construire des rapports ou bien des tableaux de bord pour les utilisateurs suivant des canevas prédéfinis par les équipes en place dans l'entreprise et qui seront reliés aux cubes présents dans le système informatique de l'entreprise.

Ces solutions tel que Microsoft® SQL Server Analysis Services (SSAS) permettent depuis une source de données de définir un Cube avec en point central les faits et des dimensions hiérarchisées. Le moyen de manipuler ces cubes peut se faire à partir d'un tableur avec une connexion au cube et des tableaux croisés dynamiques, mais aussi via des applications. Ces solutions sont le plus souvent réservées à des réalisations spécifiques et souvent professionnelles du fait de leur prix, de l'infrastructure à mettre en place et de la complexité liée à leurs réalisations.

Choisir une solution professionnelle est non seulement plus lourde en terme de conception et d'infrastructures mais également plus couteuse en terme de licences d'utilisation. Dans ce mémoire, l'angle qui a été choisi est l'usage domestique, accessible à des particuliers et des très petites entreprises. Or nous allons voir dans la section suivante comment le tableur Excel a pu évoluer pour prendre en compte la résolution de problèmes à plusieurs dimensions.

4.3 Excel, PowerPivot et le langage DAX

4.3.1 Historique

Conceptualisée dès les années 70 par René PARDO et Remy LANDAU, l'idée de feuilles de calcul a été développée en 1979 par Dan BRICKLIN sous le nom de VisiCalc pour l'Apple II. Ce premier logiciel a rapidement donné naissance à un autre logiciel de calcul qui fut 1-2-3 de Lotus pour le PC d'IBM en 1983. Ensuite d'autres évolutions et logiciels ont permis d'arriver aux tableurs actuels avec l'ajout de fonctionnalités.

Ces tableurs ont pour but d'apporter à un utilisateur une solution de résolution de problèmes de calculs sans devoir investir dans des solutions professionnelles lourdes et couteuses. Le but est de fournir à l'utilisateur un moyen d'encoder des données et, après différents calculs, de recevoir des résultats. Le fonctionnement des tableurs se base sur un principe de feuilles de calcul où l'ensemble des données et des formules de calcul s'insèrent dans des cellules situées à l'intersection de lignes et colonnes.

4.3.2 Focus sur Microsoft® Excel

Inventé en 1985 pour les ordinateurs Apple Macintosh, puis en 1987 sur Windows, Microsoft® avait déjà commercialisé un tableur « Multiplan » en 1982. C'est en 1990 avec Windows 3.0 que les ventes d'Excel sont supérieures à Lotus 1-2-3 et permettent à l'entreprise de prendre la première place et prouver à la concurrence son aptitude à développer des solutions à interface graphique utilisateur.

Lors de l'ouverture du tableur Excel, l'utilisateur se retrouve face à une fenêtre nommée feuille et composée de cellules agencées en lignes et colonnes. Lorsque plusieurs feuilles sont regroupées dans le tableur, on parle de classeur. Dans une feuille de calcul, l'utilisateur peut encoder dans des cellules son problème de calcul. Une cellule peut contenir différents types d'informations : soit une information textuelle pour décrire l'information, soit une donnée numérique ou logique qui va servir d'entrée pour la résolution du problème à résoudre, ou bien une formule qui à partir de données présentes dans d'autres cellules, va effectuer un calcul et fournir un résultat.

Excel permet de gérer différents types de données. Celles-ci peuvent être numériques, booléennes, des chaines de caractères, une date ou un temps. Chaque type de donnée est accompagné d'un ensemble de formules permet-

tant leur manipulation. Afin de marquer la distinction entre une donnée brute et une formule, le tableur Excel impose d'utiliser le signe "=" comme premier caractère à encoder dans la cellule.

Les formules dans Excel utilisent les cellules contenant les données en se basant sur leur adresse. Ces adresses représentent le positionnement de la cellule à l'intérieur de la feuille de calcul. Ce positionnement est le résultat du croisement entre la ligne et la colonne où est positionné la cellule. Les formules utilisent les adresses des cellules selon trois manières : la valeur relative, la valeur absolue ou le nom de la cellule. Référencer une cellule par valeur relative est la manière par défaut utilisé par Excel. C'est la façon de référencer directement la position ligne et colonne de la cellule. Excel gère cette position de cellule par rapport au chemin à parcourir pour l'atteindre.

4.3.3 PowerPivot

PowerPivot pour Excel est un outil d'analyse de données apparu en tant que complément dans Excel 2010 puis intégré totalement au tableur depuis Excel 2013. PowerPivot est une version locale dans un classeur Excel d'une instance de Microsoft® SQL Serveur Analysis Service évoqué dans la section 4.2.

Il permet d'importer de grandes quantités de données venant de différentes sources dans un modèle de données propre à PowerPivot afin de créer des relations entre les données importées. Le modèle de données généré en interne par PowerPivot est chargé en mémoire vive sur l'ordinateur client grâce au moteur de compression VertiPaq. Le résultat final est une nouvelle source de données incorporée dans le classeur qui sert de base pour les rapports de tableaux croisés dynamiques.

PowerPivot utilise des formules écrites en langage DAX (Data Analysis Expressions) afin de créer des mesures calculées. L'avantage de PowerPivot par rapport à Excel est qu'il est possible de créer un modèle de données plus complexe qu'une simple feuille de calcul.

Voici quelques différences entre Excel et PowerPivot :

- dans Excel les tables peuvent être regroupées dans une même feuille de calcul tandis que ces tables sont organisées en pages à onglets individuels dans la fenêtre PowerPivot ;
- Excel permet de modifier les valeurs dans une table alors que ceci n'est pas permis dans PowerPivot ;
- les calculs sont réalisés via des formules dans Excel tandis que PowerPivot

utilise le langage DAX ;

- il est possible dans PowerPivot de créer des hiérarchies dimensionnelles, des indicateurs de performances clés (KPI) et des perspectives afin de limiter le nombre de colonnes et de tables pour l'utilisateur dans sa feuille de calcul.

PowerPivot est capable de prendre en charge des fichiers jusqu'à 2Go et permet d'utiliser jusqu'à 4Go de données en mémoire.

4.3.4 Le langage DAX

Le langage DAX (Data Analysis Expressions) est un langage de formules permettant de définir des calculs personnalisés par rapport au modèle de données contenu dans PowerPivot. Ces formules sont comparables aux formules présentes dans Excel. Elles permettent de manipuler des données numériques, travailler sur des chaines de caractères, sur des dates et des heures ou bien de créer des valeurs conditionnelles.

Le contexte d'une formule DAX

Les formules DAX sont toujours évaluées en fonction du contexte dans lequel elles s'appliquent. Le contexte permet de réaliser des analyses dynamiques dans lesquelles les résultats d'une formule peuvent changer suivant que l'on se trouve sur une ligne, une sélection de cellules ou bien plusieurs données reliées entre-elles.

Il existe différents types de contextes dans les formules DAX : le contexte de ligne, le contexte de requête et le contexte de filtre. Le contexte de ligne peut être considéré comme un contexte de "ligne courante". Si on crée une colonne calculée, la formule écrite dans la colonne va s'appliquer pour toutes les lignes de la table.

Le contexte de requête correspond au sous-ensemble de données résultant des en-têtes de lignes et de colonnes, des filtres et des segments choisis dans le tableau croisé dynamique. Si on choisit de visualiser le montant des ventes réalisées par l'entreprise et que l'on choisit en en-tête de colonne l'attribut *Année*, le contexte de requête va porter sur le sous-ensemble des ventes qui ont été réalisées par années.

Le contexte de filtre est présent lorsque dans la rédaction d'une formule, celle-ci permet dans ces arguments d'ajouter un filtre. Ce contexte s'applique

toujours en priorité et au dessus des autres contextes. Par exemple la syntaxe de la formule CALCULATE est CALCULATE(<expression>,<filter1>,<filter2>...). filter1 et filter2 sont des expressions booléennes qui permettent de définir des filtres afin de réduire l'ensemble des données qui serviront au calcul de la formule.

Le contexte d'une formule DAX est déterminé à l'aide des tables dans le classeur, des relations entre ces tables et des filtres qui sont appliqués. Le contexte peut compliquer la résolution d'erreurs générées par certaines formules. C'est pourquoi de manière générale, il est recommandé d'écrire des formules simples.

Les différences entre les formules Excel et DAX sont :

- de nombreuses fonctions DAX ont le même nom et le même comportement général que les fonctions Excel, mais ont été modifiées pour accepter différents types d'entrées et, dans certains cas, peuvent retourner un type de données différent. En général, il n'est pas possible d'utiliser des formules DAX dans un classeur Excel ni d'utiliser des formules Excel dans un classeur PowerPivot sans effectuer quelques modifications ;
- une fonction DAX se réfère toujours à une table ou une colonne complète et non pas à une valeur de la table en particulier. L'utilisation de filtres peut être ajoutée si on souhaite réduire les valeurs visées par la formule. Quand on écrit une formule en Excel, les paramètres utilisés font référence à une cellule particulière dans la feuille de calcul, ou bien à une plage de données. Par contre, en DAX, une formule qui retournera le produit des quantités vendues par rapport au prix unitaire de l'article acheté va calculer un résultat pour toutes les lignes de la table des ventes.
- DAX permet de retourner une table comme résultat au lieu d'une valeur unique. Ce type de fonction peut être utile pour fournir une entrée à d'autres fonctions et permettre de calculer des valeurs sur des tables ou des colonnes ;
- DAX fournit des fonctions de recherche, similaires aux fonctions de recherche de tableau et de vecteur dans Excel. Cependant, les fonctions DAX nécessitent d'établir une relation entre les tables sur lesquelles les fonctions vont porter.

Il est possible d'imbriquer des fonctions DAX entre elles, ce qui signifie que les résultats d'une fonction sont utilisés comme un argument d'une autre fonction. L'imbrication de fonctions peut se faire jusqu'à 64 niveaux de fonctions dans les colonnes calculées. Cependant, l'imbrication peut rendre plus complexe la création ou le dépannage d'une fonction.

De nombreuses fonctions PowerPivot sont conçues pour être utilisées uniquement comme fonctions imbriquées. Ces fonctions retournent une table, qui ne peut pas être enregistrée directement comme résultat dans le modèle de données PowerPivot ; elle doit être fournie comme entrée à une fonction de table. En examinant la syntaxe de la fonction FILTER(<table>,<filter>), celle-ci demande en arguments une table contenant les données qu'il sera nécessaire de filtrer et un argument *filter* qui est une expression booléenne.

4.4 Tableaux croisés dynamiques

Les tableaux croisés dynamiques (abrégé en TCD) ou « Pivot Table » en anglais permettent d'analyser, exploiter et présenter les données d'une feuille de calcul ou d'une source de données externe. Les tableaux croisés dynamiques sont utiles quand le volume de données à exploiter est conséquent et qu'elles devront être présentées selon plusieurs axes d'observations. Le tableau est dynamique car à chaque changement dans l'organisation des lignes et colonnes du tableau, une nouvelle requête est envoyée afin de récupérer les données et de les présenter. Les TCD sont disponibles depuis la version Excel97.

Une des caractéristiques de l'analyse multidimensionnelle est de pouvoir présenter les données selon des vues et suivant plusieurs niveaux de détails. Ces vues seront construites en sélectionnant parmi les attributs de dimensions ceux qui doivent apparaitre en ligne et en colonne dans le tableau. D'autres dimensions serviront également de filtre. Une fois la vue construite, le tableau dynamique croisé va créer une requête qu'il va envoyer au modèle de données PowerPivot afin de récupérer les données correspondantes à la requête. Les tableaux croisés dynamiques vont être utiles pour la présentation de ces données, notamment en proposant l'utilisation d'un jeu d'opérations OLAP simple.

- l'opération **roll-up** est une opération qui va agréger les données suivant une dimension ou une hiérarchie. Si par exemple les points de ventes de l'entreprise sont regroupés dans une hiérarchie *magasin>pays*, grâce à l'opération roll-up, il est possible d'obtenir le montant des ventes de l'ensemble des magasins regroupés et agrégés par pays ;
- l'opération **drill-down** est à l'opposé de l'opération de roll-up. Le drill-down permet de parcourir la dimension ou la hiérarchie à un niveau de granularité plus fin. Si nous avons les données présentées par trimestre dans la dimensions date, le drill-down va permettre de visualiser les données mois par mois ;

- l'opération **pivot** est une opération de rotation des axes du cube afin de proposer une autre visualisation des données ;
- le **slice** est une opération qui va découper le cube suivant une dimension ou une hiérarchie afin d'obtenir un *sous-cube* telle une tranche qui aurait été découpée dans le cube. Citons par exemple la possibilité de visualiser les informations pour les points de ventes d'un pays uniquement ;
- le **dice** est similaire au slice sauf que la découpe du cube en un sous-cube ne va pas se faire selon une seule dimension mais sur plusieurs dimensions ;
- l'opération **drill-across** consiste à exécuter une requête impliquant plus d'un cube. La condition pour exécuter cette opération étant que les deux cubes à interroger doivent partager au moins une dimension en commun ;
- l'opération **drill-through** permet d'accéder au niveau de détail le plus fin des données stockées dans le cube.

4.5 En résumé

L'informatique décisionnelle, ou Business Intelligence, prend une part de plus en plus importante dans les entreprises cherchant à mieux exploiter et analyser les données qu'elles traitent. Pour ce faire, de plus en plus de solutions logicielles sont proposées sur le marché.

Le tableur Excel n'est pas en reste pour fournir à ces utilisateurs des outils de business intelligence. Que ce soit les tableaux croisés dynamiques dès 1997 ou bien PowerPivot et son langage DAX en 2010, l'ajout de ces outils permet à tout un chacun de développer des solutions d'analyses basées sur de grandes quantités de données venant notamment de classeurs Excel.

Conclusion

Au terme de cette partie, nous avons pu voir comment il est possible de modéliser un entrepôt de données suivant la méthode décrite dans [Kimball and Ross, 2013]. Cette méthode est pour moi une bonne manière de modéliser un petit entrepôt de données afin de répondre aux besoins du public cible de ce mémoire. Il n'y a pas de grandes transformations à mettre en place tel que le passage par des étapes de modèles conceptuel, logique et physique comme on peut les retrouver lors de modélisation de base de données relationnelles. De même, il n'est pas nécessaire d'utiliser une notation spécifique propre au modèle. Le résultat obtenu par ce modèle dimensionnel va être un schéma en étoile qui reprendra pour chaque table les attributs qui la composent. Ce modèle sera facilement lisible et compréhensible par un utilisateur sans expérience informatique.

D'autre part, nous avons pu voir une modélisation d'un problème de calcul créée par Jean-Luc HAINAUT qui permet de définir des règles qui seront à implémenter dans un tableur afin de pouvoir résoudre les problèmes posés. Ce modèle, bien que très abstrait et conceptuel, est également compréhensible pour des utilisateurs ayant une expérience dans les tableurs sans pour autant nécessiter des compétences en développement informatique.

Enfin, nous avons abordé les outils qui vont être utilisés pour la mise en place d'une solution. Nous nous sommes particulièrement concentrés sur le tableur Excel de chez Microsoft® qui dispose d'outils complémentaires permettant de créer un entrepôt de données, de définir dans le langage DAX des mesures calculées qui sont le résultat de calculs sur les données en entrée et enfin de proposer un outil de visualisation de ces résultats dans une feuille du classeur grâce aux tableaux croisés dynamiques.

Deuxième partie

Concevoir et implémenter un modèle dimensionnel dans un contexte domestique

Introduction

Après avoir vu dans la première partie de ce mémoire les différentes modélisations ainsi que des outils utiles afin de résoudre un problème de calcul à plusieurs dimensions, nous allons dans cette partie détailler les différentes étapes de transformations nécessaires afin d'obtenir une implémentation dans Excel d'un problème de calcul à plusieurs dimensions.

La modélisation dimensionnelle telle que décrite dans [Kimball and Ross, 2013] met l'accent sur l'identification des dimensions et des faits sans pour autant décrire comment les exigences des utilisateurs doivent se refléter dans les mesures calculées. Il en est de même dans [Malinowski and Zimányi, 2008] où la manière de traiter les mesures calculées n'est que peu abordée. Le risque étant qu'avec des mesures calculées mal définies, la résolution du problème risque de ne pas rencontrer son objectif.

L'idée principale va être dans une première étape de considérer le modèle dimensionnel de Ralph KIMBALL comme un modèle des données de l'entrepôt de données sur la base des données collectées par l'entreprise. Lors d'une seconde étape, le modèle de Jean-Luc HAINAUT va servir pour répondre aux exigences des utilisateurs et va permettre d'identifier les règles de calcul à mettre en œuvre afin d'obtenir les résultats souhaités et ainsi définir les mesures calculées de l'entrepôt de données. Au terme de ces deux étapes, les mesures identifiées dans le modèle Hainaut seront ajoutées au modèle dimensionnel pour aboutir à un modèle d'entrepôt de données prêt à être implémenté dans le tableur Excel.

Finalement nous aborderons l'implémentation dans un tableur domestique, en l'occurrence Microsoft® Excel 2016, en utilisant les outils PowerPivot et le language DAX qui lui est associé ainsi que les tableaux croisés dynamiques pour visualiser les résultats.

Chapitre 5

Méthodologie de création d'un modèle dimensionnel adapté aux usages domestiques

5.1 Introduction

Dans ce chapitre, nous allons voir comment créer un modèle dimensionnel basé sur la méthodologie de Ralph KIMBALL mais limité aux mesures factuelles identifiées lors du processus de conception du modèle. La notation utilisée pour la réalisation du modèle dimensionnel sera légèrement différente de la notation utilisée habituellement afin de pouvoir servir à l'élaboration du modèle Hainaut par la suite.

Dans un deuxième temps, nous allons réaliser le modèle Hainaut qui va nous permettre de définir les résultats ainsi que les règles permettant de répondre aux exigences des utilisateurs. Ensuite, nous allons reprendre le modèle dimensionnel afin d'y inclure les mesures calculées qui ont été définies dans le modèle Hainaut et d'y ajouter les clés permettant de lier les différentes tables du modèle.

5.2 Réalisation d'un schéma en étoile

Après avoir suivi les différentes étapes permettant d'identifier les faits et les dimensions suivant la méthode décrite dans [Kimball and Ross, 2013] au chapitre 2, un schéma en étoile va pouvoir être construit. Le but de ce schéma va

être de visualiser l'organisation des tables de faits et les dimensions afin de valider le modèle en s'assurant que les tables de faits du modèle sont bien reliées aux dimensions auxquelles elles doivent se rapporter.

Le schéma en étoile qui va être construit ici sera différent dans sa notation par rapport au schéma résultant de la méthodologie de Ralph KIMBALL tel qu'on a pu le voir dans la figure 2.4 à la fin du chapitre 2. Cette proposition de notation pour la conception du modèle dimensionnel est volontairement semblable à la notation utilisée dans PowerPivot afin qu'au terme de l'implémentation du modèle, l'utilisateur puisse comparer son modèle avec celui qui est généré par PowerPivot. Cette notation qui inclut également les types des données présentes dans l'entrepôt va permettre, lors de l'élaboration du modèle Hainaut, de vérifier la cohérence entre les données du modèle.

En premier lieu, on va placer au centre du schéma les tables de faits en les identifiant par leur nom. Ces tables de faits vont être enrichies par les mesures se rapportant aux faits telles qu'elles ont été définies dans la section 2.5. On va ajouter également dans le schéma les dimensions identifiées dans la section 2.4 avec leurs attributs.

Ensuite, pour tous les attributs et mesures inclus dans une des tables du schéma, on va ajouter le type de donnée auquel ils se rapportent. Cette étape va permettre, lors de la création du modèle Hainaut, de vérifier que les types de données en rapports aux mesures ou attributs sont cohérents.

Date	
DateComplète	Date
JourDuMois	Entier
NuméroMois	Entier
NomMois	Texte
AnnéeCalendrier	Entier
TrimestreCalendrier	Entier
AnnéeFiscale	Entier
TrimestreFiscal	Entier
Année-Mois-Jour	
AnnéeCalendrier NomMois JourDuMois	

FIGURE 5.1 – Exemple de notation des hiérarchies dans le schéma en étoile

Les hiérarchies dimensionnelles seront représentées dans le schéma dans les dimensions auxquelles elles se rapportent. Pour chaque hiérarchie, on reportera son nom à la fin de la liste des attributs de la dimension et ensuite on

listera les attributs de la hiérarchie les uns sous les autres en respectant le lien hiérarchique entre les attributs. Par exemple, dans la dimension Date, la hiérarchie *Année-Mois-Jour* sera reportée comme sur la figure 5.1.

Le résultat produit sera un schéma en étoile semblable à la figure 5.2 et qui va nous guider dans la conception du modèle Hainaut.

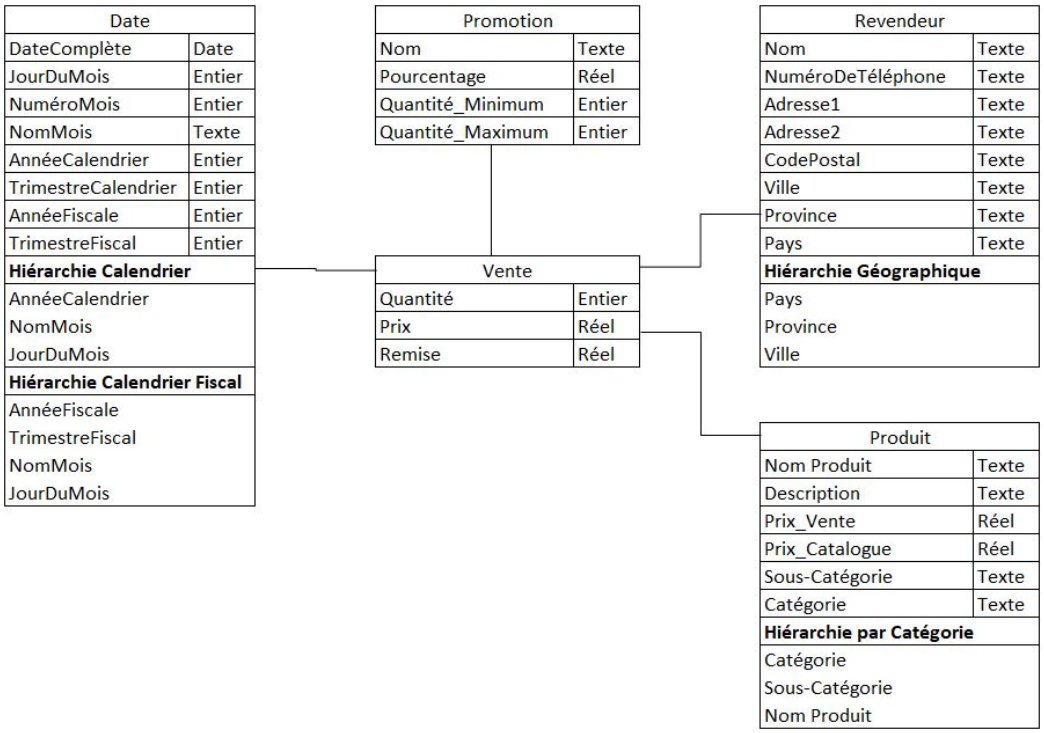


FIGURE 5.2 – Schéma en étoile résultant de la méthode KIMBALL

5.3 Réalisation du modèle Hainaut

La réalisation du modèle Hainaut de la problématique va reprendre la méthode qui a été évoquée dans le chapitre 3. Ce modèle va reprendre en tant que résultat les exigences utilisateurs et définir les règles permettant d'obtenir ces résultats à partir des données présentes dans l'entrepôt de données.

Par exemple, si on souhaite interroger l'entrepôt de données afin d'obtenir les montants des ventes par produits vendus ainsi que la part de marché que chaque produit représente, le modèle va ressembler à ceci :

Résultats

$\text{MONTANT}_{D,Prom,R}$: réel (€) ; Montant des ventes

$\text{PART\_DE\_MARCHÉ}_{D,P,Prom,R}$: réel (%) ; Part de marché par produit

Grandeurs internes

$\text{SOUS\_MONTANT}_{D,P,Prom,R}$: réel (€) ; Montant des articles à payer sans prendre en compte la remise

$\text{MONTANT\_REMISE}_{D,P,Prom,R}$: réel (€) ; Montant de la remise à déduire

$\text{MONTANT}_{D,P,Prom,R}$: réel (€) ; Montant à payer pour un produit déduit du montant de la remise accordée à ce produit

Règles

$\text{MONTANT}_{D,Prom,R} = \sum_p \text{MONTANT}_{D,P,Prom,R}$

$\text{MONTANT}_{D,P,Prom,R} = \text{SOUS\_MONTANT}_{D,P,Prom,R} - \text{MONTANT\_REMISE}_{D,P,Prom,R}$

$\text{SOUS\_MONTANT}_{D,P,Prom,R} = \text{PRIX}_{D,P,Prom,R} * \text{QUANTITÉ}_{D,P,Prom,R}$

$\text{MONTANT\_REMISE}_{D,P,Prom,R} = \text{SOUS\_MONTANT}_{D,P,Prom,R} * (1-\text{REMISE}_{D,P,Prom,R})$

$\text{PART\_DE\_MARCHÉ}_{D,P,Prom,R} = \text{MONTANT}_{D,P,Prom,R} / \text{MONTANT}_{D,Prom,R}$

Dans ce modèle Hainaut, les grandeurs sont toutes dimensionnées par l'ensemble des dimensions qui identifient un fait. La notation du modèle Hainaut implique que les dimensions qui caractérisent une grandeur doit apparaitre en indice de cette grandeur. Certaines grandeurs pouvant être indicées par de nombreuses dimensions, il est permis d'utiliser une abréviation de ces dimensions en tant qu'indice afin de garder le texte clair auprès de l'utilisateur. C'est pourquoi lorsque l'on retrouve dans le modèle une grandeur telle que $\text{MONTANT}_{D,P,Prom,R}$, cela signifie que la valeur représente le montant d'un fait "Vente" pour lequel une date(D), un produit(P), une promotion(Prom) et un revendeur(R) ont été sélectionnés. Grâce aux tableaux croisés dynamiques et à PowerPivot, nous verrons dans le prochain chapitre que si la sélection des membres de dimensions retournent plusieurs faits, le montant représentera une agrégation automatique des montants et non plus un seul et unique montant correspondant à un et un seul fait. Par exemple si nous choisissons dans la dimension Date non pas une date mais un mois ou une année, la valeur de $\text{MONTANT}_{D,P,Prom,R}$ représentera le montant des ventes pour un produit, une promotion, un revendeur, mais pour plusieurs dates.

5.4 Retour vers le modèle dimensionnel

Après avoir réalisé le modèle Hainaut du problème à résoudre, le modèle dimensionnel doit être mis à jour afin de prendre en compte les nouvelles me-

sures calculées que le modèle Hainaut a produit. Pour réaliser cela, on va reprendre le schéma en étoile qui a été produit dans la section 5.2 et on va ajouter dans ce schéma toutes les grandeurs de type résultat ou intermédiaire résultant du modèle Hainaut.

Afin de respecter le troisième principe énoncé par Jean-Luc HAINAUT dans la section 3.2, on va mettre à jour le schéma en étoile afin de retirer toutes les mesures factuelles n'intervenant pas dans les règles élaborées lors de la conception du modèle Hainaut. L'avantage sera de ne garder que des mesures qui seront réellement utilisées pour la résolution du problème énoncé.

Le schéma va donc maintenant contenir l'ensemble des tables de faits avec leurs mesures ainsi que les dimensions et leurs attributs. Il reste à définir les relations entre les tables de faits et les dimensions qui s'y rapportent. Toutes les dimensions vont contenir une clé primaire permettant d'identifier de manière unique les membres des dimensions. Les tables de faits vont, elles, contenir un ensemble de clés étrangères relatives aux dimensions qui sont liées aux tables de faits. Le résultat de cette dernière transformation va fournir un schéma en étoile (Figure 5.3) qui va pouvoir par la suite être implémenté dans le tableur Excel.

5.5 En résumé

Dans ce chapitre, nous avons pu voir une méthode permettant de réaliser un modèle dimensionnel qui combine deux modélisations. La modélisation dimensionnelle suivant la méthode de Ralph KIMBALL a permis d'identifier les faits et les dimensions de la problématique en créant un modèle d'entrepôt de données. Ce modèle dimensionnel a été modifié dans sa notation afin d'être plus facilement compréhensible, mais également en y incluant les types des données qui seront ensuite utiles lors de la réalisation du modèle Hainaut.

Le modèle Hainaut est quant à lui réalisé selon les principes énoncés dans [Hainaut, 2002] exception faite qu'il n'est plus nécessaire de lister l'ensemble des grandeurs en entrée qui seront utilisées dans le modèle Hainaut puisque celles-ci sont déjà présentes dans le modèle dimensionnel réalisé plus tôt.

Finalement, toutes les grandeurs internes et les grandeurs de type résultat, résultantes du modèle Hainaut, sont incorporées aux mesures du modèle dimensionnel et les identifiants des dimensions et des faits sont ajoutés.

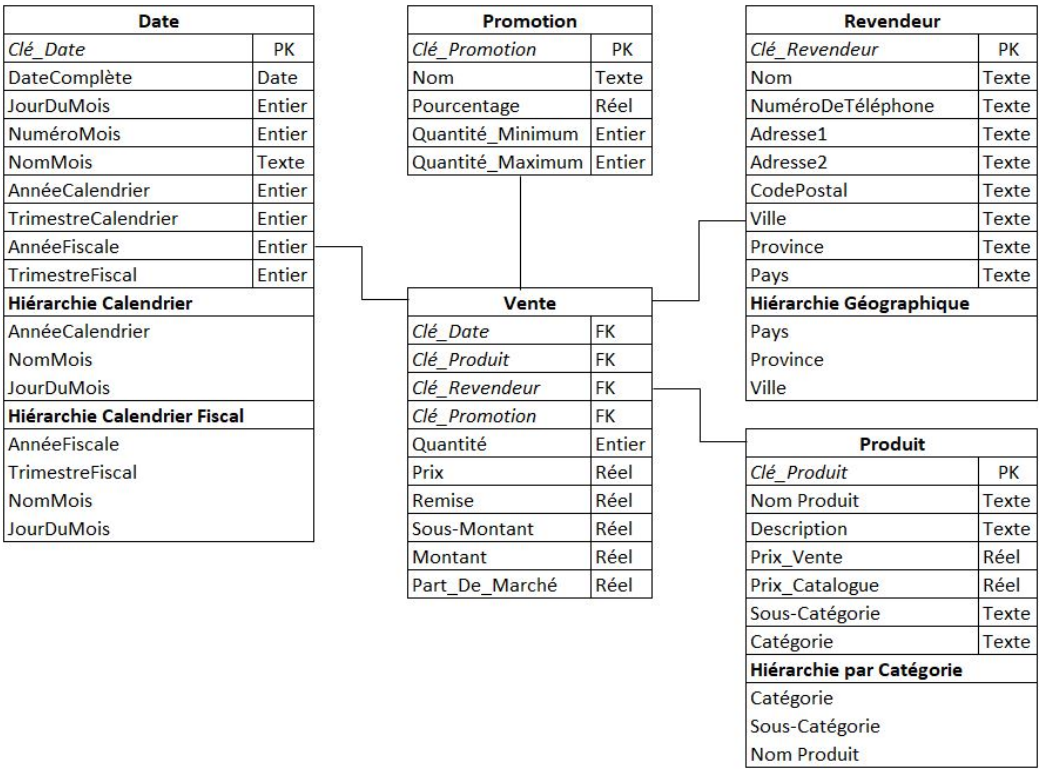


FIGURE 5.3 – Schéma en étoile enrichi des mesures calculées et des relations

Chapitre 6

Implémentation d'un modèle dimensionnel dans Excel

6.1 Introduction

Après avoir réalisé le modèle dimensionnel et le modèle Hainaut, nous devons les traduire en une implémentation dans le tableur Excel. Dans ce chapitre, nous allons en premier lieu créer les tableaux qui contiendront les données de notre entrepôt de données sur la base du modèle dimensionnel. Ces tableaux seront importé dans l'outil Power Pivot d'Excel afin d'obtenir un modèle de données Excel. Ensuite nous transformerons les règles de calcul définies lors de la conception du modèle Hainaut en mesures ou en colonnes calculées à l'aide du langage DAX. Finalement, nous aborderons la manière de présenter les résultats dans un tableau croisé dynamique dans le classeur Excel qui pourra être manipulé par l'utilisateur.

C'est bien la combinaison des deux modèles réalisés qui vont permettre d'obtenir une implémentation dans Excel sous la forme d'un modèle de données PowerPivot dans le classeur.

6.2 Créer les tableaux Excel

Chaque dimension et chaque table de faits du modèle dimensionnel seront reportées en tant que tableaux dans Excel sous une forme aplatie. Les dimensions comprenant des hiérarchies seront elles aussi reportées sous une forme aplatie quitte à répéter les même valeurs dans les attributs de membres ap-

partenant au même ensemble dans la hiérarchie tels qu'on peut le voir dans la figure 2.3 dans le chapitre 2.

Dans le cadre de notre approche transformationnelle, chaque tableau Excel portera le nom donné à la dimension ou à la table de fait du modèle et chaque attribut ou mesure sera défini comme une colonne du tableau. Chaque enregistrement sera donc repris en tant qu'une ligne de ces tableaux. La procédure pour créer un tableau dans Excel 2016 est reprise dans [Lemainque, 2016] pages 69 à 71. Pour résumer, sélectionnez l'ensemble des données et des entêtes de la table à transformer en tableaux et dans le ruban *Insérer* cliquez sur l'icône *Tableau*. Il existe un raccourci clavier permettant d'obtenir le même résultat, **CTRL+L**. Dans la boite de dialogue, vérifiez que la plage de données corresponde à votre sélection, que la case "Mon tableau comporte des en-têtes" est bien cochée et cliquez sur le bouton "OK" (voir figure 6.1). Finalement, donnez le nom de l'entité au tableau nouvellement créé.

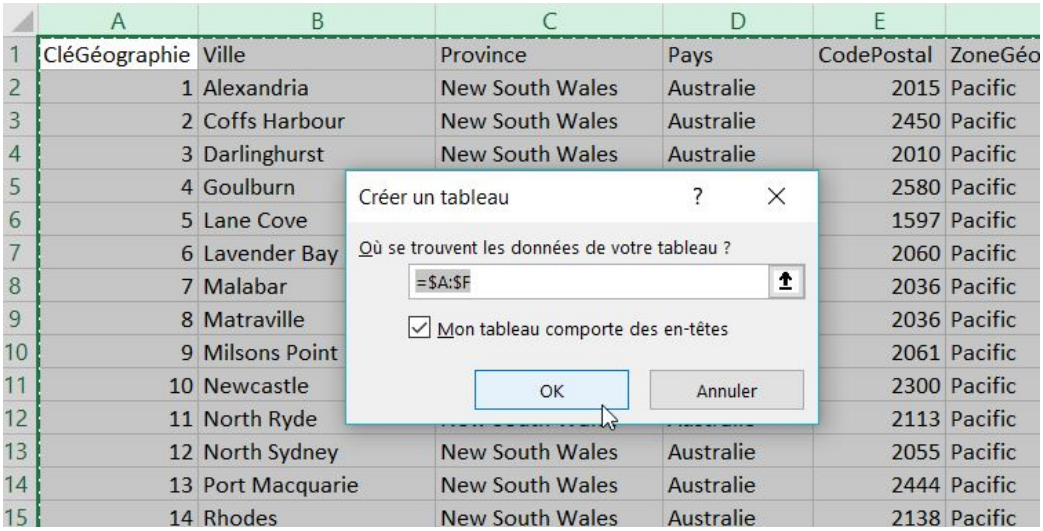


FIGURE 6.1 – Création d'un tableau en Excel

Les données qui vont garnir ces tableaux proviendront soit des données générées par les systèmes opérationnels ou bien pourront être introduites manuellement dans le tableaux.

6.3 Importer les données dans PowerPivot

Une fois l'étape de création des tableaux dans Excel 2016 réalisée, il faut importer les données dans PowerPivot. PowerPivot accepte de nombreuses sources de données dont celles incluses dans un ou plusieurs classeurs Excel.

Importer les différents tableaux créés à l'étape précédente ne représente pas une grande difficulté. Il suffit de se positionner sur une cellule du tableau à importer et de cliquer sur l'icône "*Ajouter au modèle de données*" qui se trouve dans le ruban *Power Pivot*. Le résultat est la création d'une table liée dans le module PowerPivot du classeur Excel. En tant que table liée, l'ajout d'un enregistrement dans le tableau qui se trouve dans le classeur Excel, sera automatiquement ajouté au modèle dans PowerPivot.
Dans PowerPivot, les tableaux et les données qu'ils contiennent sont visibles dans un onglet par tableaux (voir figure 6.2).

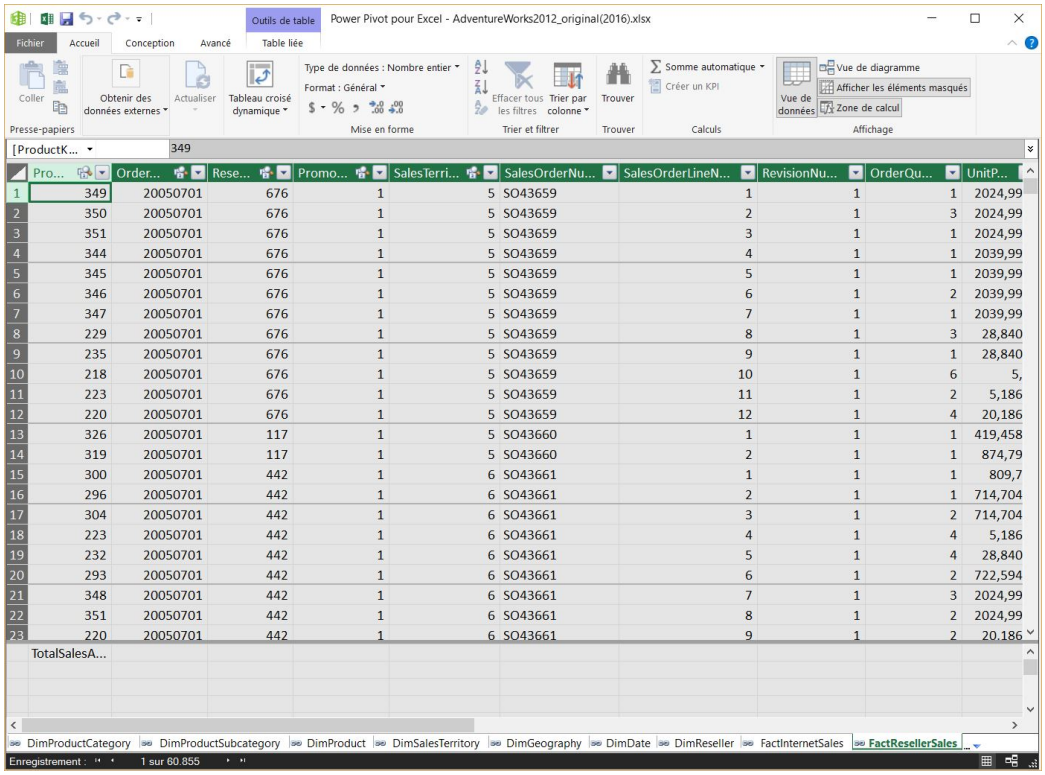


FIGURE 6.2 – Vue des données dans PowerPivot

Il est également possible d'obtenir une visualisation graphique du modèle PowerPivot telle que sur la figure 6.3. Cette visualisation est possible en cliquant

sur le bouton "Vue de diagramme" dans le ruban *Accueil*. Grâce à cette visualisation, il va être possible de faire une comparaison avec le modèle en étoile réalisé lors de la modélisation dimensionnelle dans la section 5.2. Il faudra s'assurer que les relations entre les dimensions et les tables de faits du modèle PowerPivot soient identiques au modèle dimensionnel.

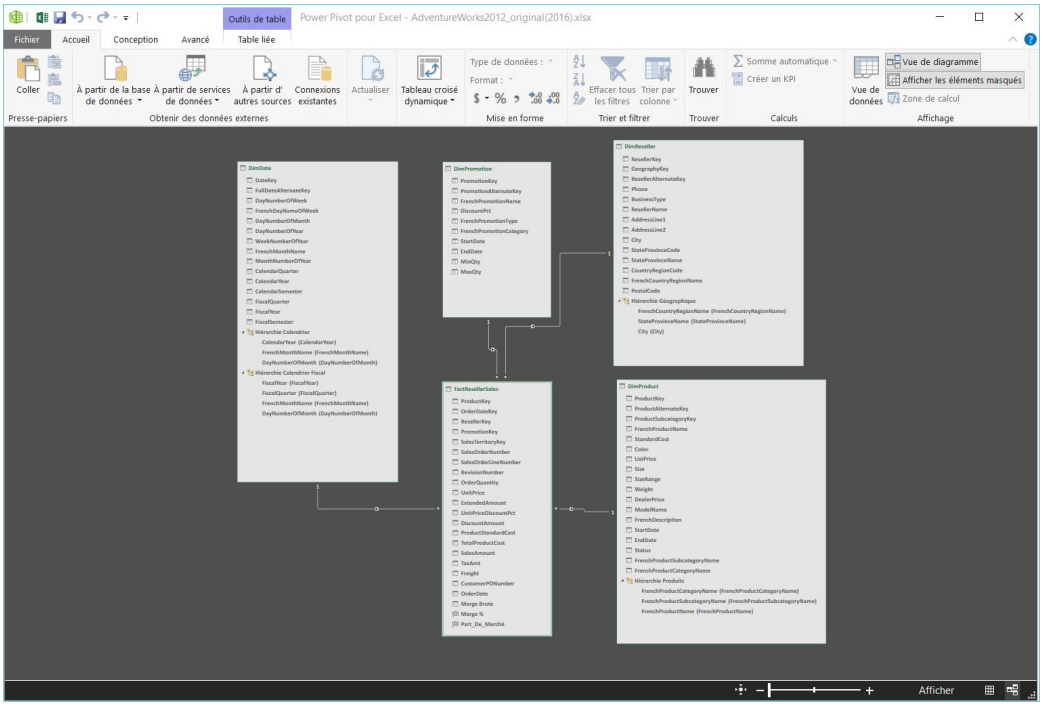


FIGURE 6.3 – Vue diagramme dans PowerPivot

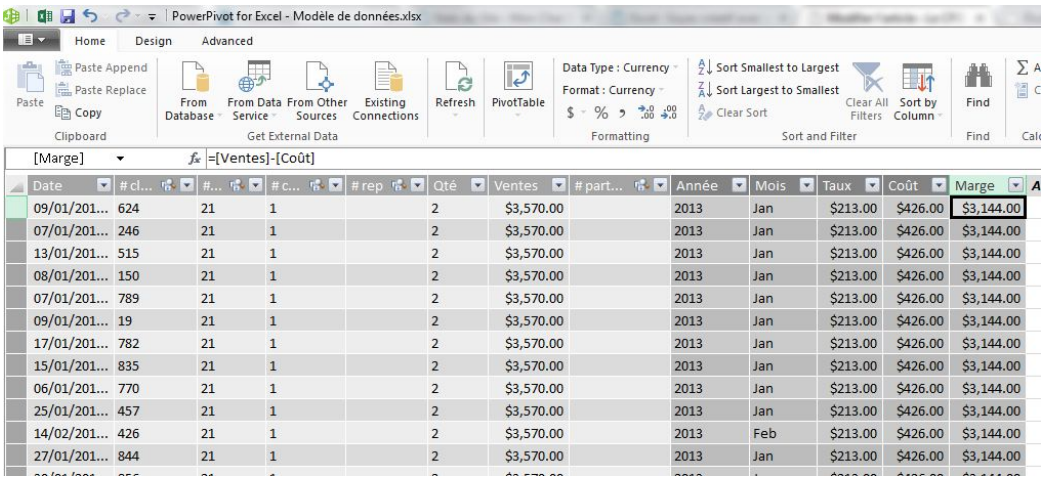
La vue diagramme du modèle PowerPivot va permettre de créer les hiérarchies dimensionnelles qui ont été définies lors de la conception du modèle dimensionnel. En sélectionnant la dimension où la hiérarchie doit être crée, cliquez sur le bouton *"Créer une hiérarchie"* situé en haut à droite. Nommez cette hiérarchie et ensuite faite glisser les attributs de la dimension qui vont composer la hiérarchie.

Au terme de ces étapes, le modèle PowerPivot créé est ainsi semblable au modèle dimensionnel qui a été défini dans la section 5.2. Il restera à ajouter les mesures et les colonnes calculées qui ont été définies dans le modèle Hainaut dans ce modèle PowerPivot afin d'obtenir un entrepôt de données fonctionnel et permettant de résoudre la problématique énoncée.

6.4 Transformer les règles en mesures et colonnes calculées

Après avoir créé le modèle PowerPivot sur la base du modèle dimensionnel, la prochaine étape va être de créer les mesures calculées correspondant aux résultats attendus. Ces mesures correspondent aux grandeurs de type résultat et aux grandeurs internes du modèle Hainaut et vont être construites dans PowerPivot à l'aide du langage DAX. Dans le chapitre 4, nous avons pu voir que le langage DAX est un langage de formules et le contenu de ces formules va être le résultat de la transposition des règles définies dans le modèle Hainaut.

On retrouve dans PowerPivot, deux sortes de résultats à une formule DAX : une **mesure calculée** ou une **colonne calculée**. Une colonne calculée est le résultat d'une formule DAX appliquée sur chaque ligne individuelle dans la table PowerPivot. En prenant en exemple le calcul de la marge brute réalisée sur les ventes, ce calcul est la différence entre le montant d'une vente et le coût de celle-ci. Ce résultat s'applique individuellement pour chaque ligne de la table des ventes et donc le résultat de ce calcul va être concrétisé par l'ajout d'une colonne calculée dans la table des ventes comme on peut le voir sur la figure 6.4.



Date	# cl...	#...	# c...	# rep	Qté	Ventes	# part...	Année	Mois	Taux	Coût	Marge
09/01/201...	624	21	1		2	$3,570.00		2013	Jan	$213.00	$426.00	$3,144.00
07/01/201...	246	21	1		2	$3,570.00		2013	Jan	$213.00	$426.00	$3,144.00
13/01/201...	515	21	1		2	$3,570.00		2013	Jan	$213.00	$426.00	$3,144.00
08/01/201...	150	21	1		2	$3,570.00		2013	Jan	$213.00	$426.00	$3,144.00
07/01/201...	789	21	1		2	$3,570.00		2013	Jan	$213.00	$426.00	$3,144.00
09/01/201...	19	21	1		2	$3,570.00		2013	Jan	$213.00	$426.00	$3,144.00
17/01/201...	782	21	1		2	$3,570.00		2013	Jan	$213.00	$426.00	$3,144.00
15/01/201...	835	21	1		2	$3,570.00		2013	Jan	$213.00	$426.00	$3,144.00
06/01/201...	770	21	1		2	$3,570.00		2013	Jan	$213.00	$426.00	$3,144.00
25/01/201...	457	21	1		2	$3,570.00		2013	Jan	$213.00	$426.00	$3,144.00
14/02/201...	426	21	1		2	$3,570.00		2013	Feb	$213.00	$426.00	$3,144.00
27/01/201...	844	21	1		2	$3,570.00		2013	Jan	$213.00	$426.00	$3,144.00

FIGURE 6.4 – Ajout d'une colonne calculée représentant la marge brute d'une ligne dans la table de ventes

Les colonnes calculées peuvent être utilisée lorsque le calcul s'applique à chaque ligne d'une table et lorsque le résultat peut par la suite être agrégé sur une dimension donnée.

Il arrive parfois que le résultat d'une colonne calculée ne puisse pas être agrégé sur une ou plusieurs dimensions. Tout comme il est parfois possible que le calcul ne puisse pas s'appliquer pour une et une seule ligne de la table mais plutôt sur une agrégation de plusieurs lignes. Il faut donc avoir recours dans ces cas aux mesures calculées dans PowerPivot. En reprenant l'exemple précédent, si l'on souhaite obtenir la marge brute en pourcentage plutôt qu'en montant, on pourrait ajouter une colonne calculée qui nous donnerait ce pourcentage pour chaque vente. Pourtant lorsque l'on souhaitera visualiser ce pourcentage en choisissant d'agréger les résultats par catégorie de produits, PowerPivot va retourner la somme des pourcentages de chaque ventes correspondantes à la catégorie de produits ce qui sera un résultat erroné puisqu'on souhaite obtenir le pourcentage sur la somme des marges.

Il est possible de créer des mesures calculées en cliquant sur le bouton *"Mesures"* qui se trouve dans le ruban *Power Pivot* du classeur Excel. Dans la fenêtre qui apparait (voir figure 6.5), choisissez la table à laquelle la mesure calculée se rapporte, définissez un nom et au besoin une description. Ensuite, écrivez votre formule DAX permettant de calculer la mesure et choisissez le formatage souhaité lors de l'affichage de la mesure calculée.

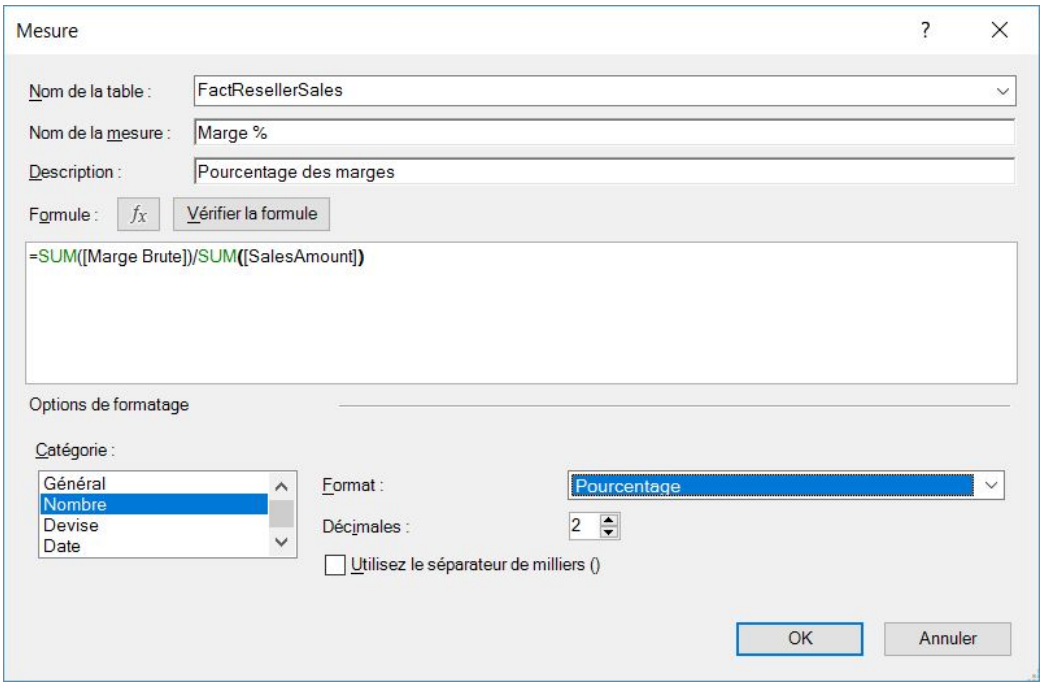


FIGURE 6.5 – Fenêtre de création d'une mesure calculée

6.5 Création des tableaux croisés dynamiques

La dernière étape dans notre approche est de présenter les données afin de répondre aux interrogations posées dans l'énoncé du problème. La première étape est de créer, dans une feuille du classeur Excel, un tableau dynamique croisé connecté au modèle créé dans Power Pivot.

Pour inclure un tableau croisé dynamique dans votre classeur Excel, sélectionnez une cellule dans une feuille du classeur et cliquez que le bouton *"Tableau croisé dynamique"* qui se trouve dans le ruban *"Insérer"*. Dans la fenêtre de création du tableau croisé dynamique, sélectionnez l'option *"Utiliser le modèle de données de ce classeur"* [Figure 6.6] qui va faire le lien entre le tableau croisé dynamique et le modèle de données qui a été créé dans PowerPivot.

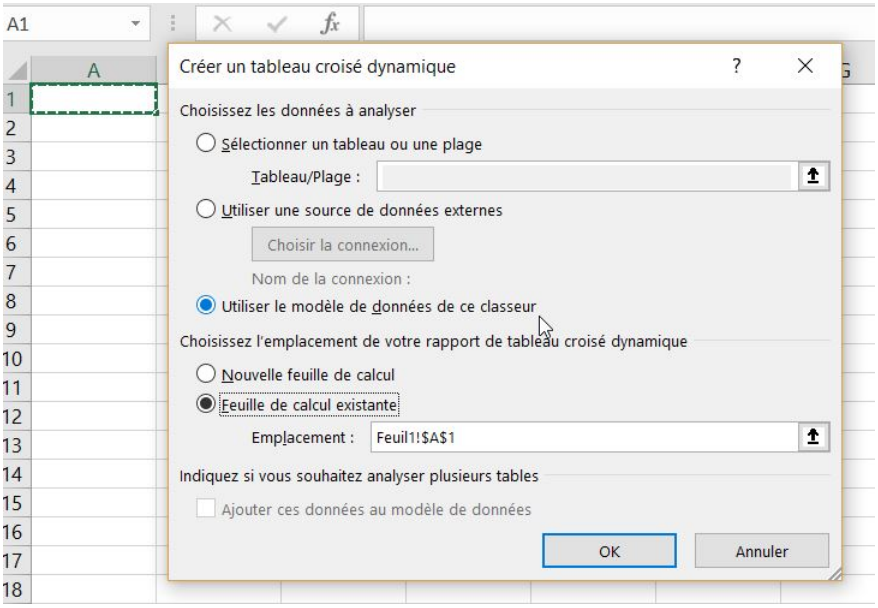


FIGURE 6.6 – Création d'un tableau croisé dynamique lié au modèle de données PowerPivot du classeur.

Le résultat de cette création de tableau croisé dynamique est visible dans la feuille du classeur et se présente sous la forme d'une fenêtre comportant l'ensemble des dimensions et leurs attributs ainsi que les tables de faits et les mesures qu'elles contiennent. De plus, cette fenêtre comporte 4 zones nommées *Filtres, Colonnes, Lignes* et *Valeurs* comme on peut le voir sur la figure 6.7. Ces zones vont permettre à l'utilisateur d'organiser les attributs et les mesures pour obtenir la présentation des résultats voulus par celui-ci en glissant

et déposant les attributs, les hiérarchies dimensionnelles et les mesures dans ces zones. Le résultat de cette manipulation sera directement visible dans le ta-

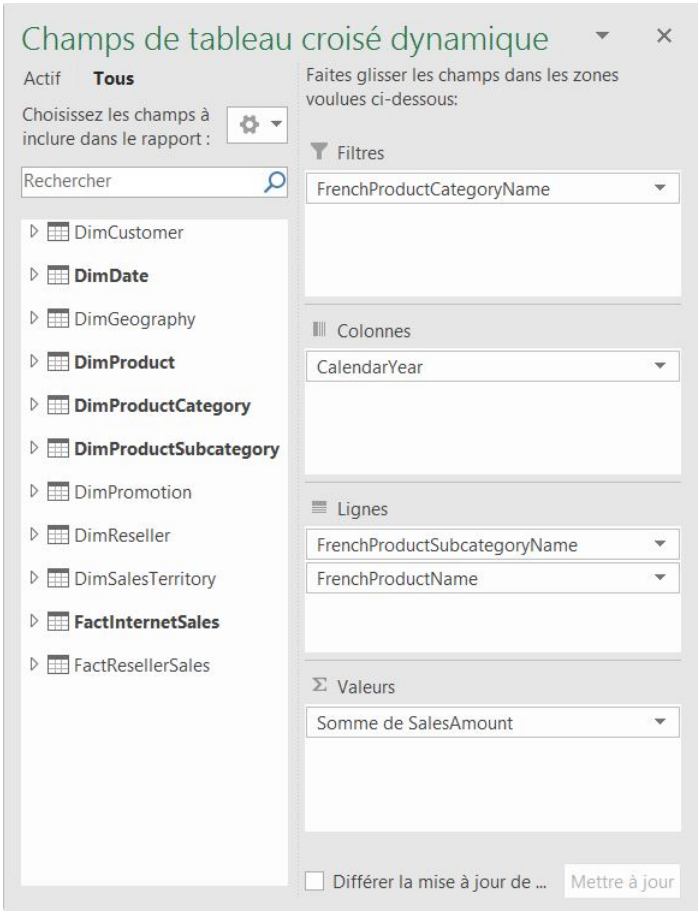


FIGURE 6.7 – Fenêtre de sélection d'un tableau croisé dynamique

bleau croisé dynamique et les résultats obtenus seront automatiquement mis à jour à chaque changement dans l'organisation du tableau.

C'est en relisant l'énoncé de la problématique à résoudre que l'utilisateur va pouvoir construire le tableau croisé dynamique en sélectionnant la ou les mesures calculées du modèle PowerPivot dans la zone *Valeurs* et ajouter les attributs de dimensions dans les zones *Colonnes* ou *Lignes* pour afficher les libellés. Si on souhaite filtrer les résultats suivants des membres de dimensions spécifiques, les attributs de dimensions pour lesquels le filtrage va devoir s'appliquer doivent être déposés dans la zone *Filtres*.

Dans les cas où l'on souhaiterait afficher en colonne ou en ligne des attri-

buts de dimensions qui devraient par ailleurs ne reprendre qu'un sous-ensemble des membres de cette dimension, il est possible de filtrer ces membres en cliquant sur la flèche qui se trouve à côté de l'attribut dans la partie de la fenêtre présentant les dimensions (voir Figures 6.8 et 6.9) et d'ensuite sélectionner les membres à afficher en cochant un ou plusieurs membres de cette dimension.

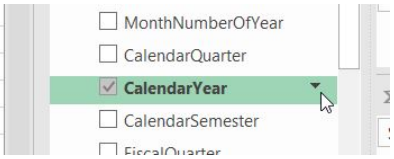


FIGURE 6.8 – Filtrer des attributs de dimensions

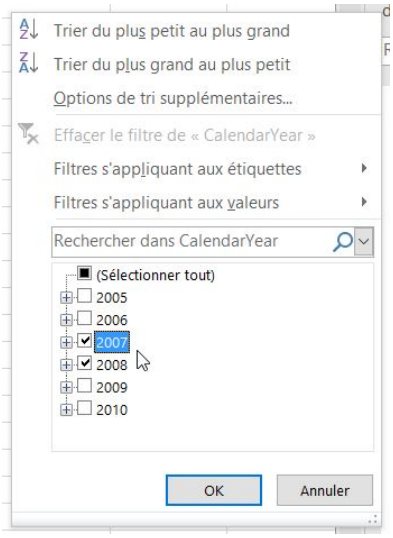


FIGURE 6.9 – Sélection des membres de dimensions à afficher

En prenant pour exemple le souhait de l'utilisateur d'afficher l'évolution des parts de marché par catégorie de produit, années après années, pour les ventes réalisées en France, nous allons détailler la création d'un tableau croisé dynamique utilisant le modèle de données PowerPivot qui a été conçu en suivant les étapes décrites dans le chapitre 5.

En premier lieu, il faut créer dans une nouvelle feuille du classeur un tableau croisé dynamique connecté au modèle de données du classeur. Ensuite, nous allons sélectionner dans la table de faits la mesure calculée "Part de marché" qui a été créée grâce aux règles de calculs du modèle Hainaut et transposée

en langage DAX dans PowerPivot. Cette mesure va être glissée et déposée dans la zone "*Valeurs*".

Puisque la demande de l'utilisateur est de visualiser ses parts de marchés années après années et par catégories de produits, sélectionnez dans la dimensions "Date" la hiérarchie Calendrier et déposez la dans le champs "*Colonnes*" et depuis la dimension "Produit", faite glisser et déposez dans le champs "*Lignes*" la hiérarchie par Catégorie. Finalement, l'énoncé demandait d'obtenir le résultat pour les ventes réalisées auprès des revendeurs français, déposez la hiérarchie Géographique de la dimension Revendeurs dans le champs "*Filtres*" et ensuite en haut du TDC, sélectionnez le membre "France" parmi la liste des pays.

Nous obtenons au final un tableau semblable à la figure 6.10 et qui permet de visualiser la réponse à la question posée par l'utilisateur.

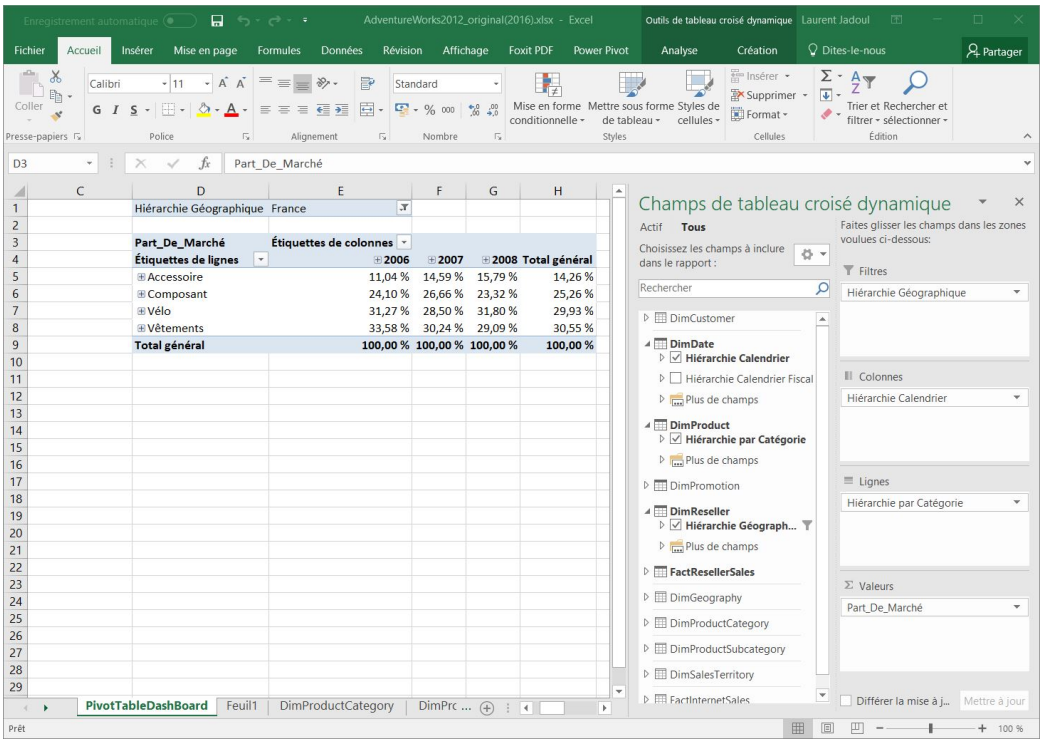


FIGURE 6.10 – TCD représentant l'évolution des parts de marchés par catégorie de produits pour les ventes réalisées en France

6.6 En résumé

Nous avons pu voir tout au long de ce chapitre une façon de transformer les modèles précédemment définis en un classeur Excel utilisant un modèle de données réalisé dans PowerPivot. La principale difficulté dans l'implémentation des deux modèles se situe au niveau de la création des mesures calculées résultant des règles définies dans le modèle Hainaut. Le choix des bonnes formules DAX va être crucial dans cette étape de la transformation.

En utilisant les tableaux croisés dynamiques, on obtient une présentation des données et des résultats qui peuvent être agencés suivant les envies de l'utilisateur. Que l'on souhaite afficher certaines données en ligne ou en colonne, le tableau croisé dynamique va interroger le modèle de données PowerPivot et retourner les informations correspondantes à la requête de l'utilisateur de manière transparente pour celui-ci et gérer la mise en forme des données dans la feuille du classeur.

Conclusion

L'objectif de ce mémoire était de trouver une méthode afin de résoudre un problème de calcul comportant des données dimensionnées. La résolution de ce type de calculs passe, la plupart du temps, par l'usage de solutions professionnelles de business intelligence alors que l'objectif était ici de rester concentré sur un public peu habitué à ce genre de solutions. Nous avons donc fait le choix de n'utiliser que les outils du tableur Excel qui sont bien connus du public cible.

Durant le développement du mémoire, l'objectif a été de mettre en place un entrepôt de données en utilisant les méthodes et les concepts permettant de créer un modèle dimensionnel tel que celui théorisé par Ralph KIMBALL. Bien que le tableur Excel permette de créer directement des tableaux croisés dynamiques sur des plages de données, implémenter un entrepôt de données a permis de définir des mesures calculées spécifiques afin de répondre aux exigences des utilisateurs. Ces mesures calculées ont pu être définies dans notre approche grâce à l'utilisation de la modélisation Hainaut qui permet de définir clairement et simplement des règles de calculs pouvant résoudre les problèmes.

La méthode qui a été présentée dans ce mémoire peut paraitre simple et en effet, des cas particuliers de hiérarchies, de dimensions ou de tables de faits spécifiques n'ont pas été abordés. Nous avons néanmoins pu développer une méthodologie qui, dans bien des cas par rapport au public visé par ce mémoire, va leur permettre de résoudre des problèmes de calculs dont les données sont dimensionnées.

A l'avenir, il serait intéressant d'évaluer la faisabilité de cette méthode avec des entrepôts de données plus complexes. Tout comme il serait opportun d'évaluer la possibilité d'automatiser la transformation des règles de calculs du modèle Hainaut vers des formules en langage DAX.

Bibliographie

[Burquier, 2007] Burquier, B. (2007). *Business Intelligence avec SQL Server 2005 : Mise en oeuvre d'un projet décisionnel*. Applications & métiers. Dunod.

[Da Costa, 2011] Da Costa, C. (2011). Mettre en place un entrepôt de données multidimensionnel. `https://business-intelligence.developpez.com/tutoriels/DWH_multidimensionnel/`.

[Hainaut, 2002] Hainaut, J.-L. (2002). *Bases de données et modèles de calcul : Outils et méthodes pour l'utilisateur Cours et exercices corrigés*. Sciences sup. Dunod, 3 edition.

[Kimball and Ross, 2013] Kimball, R. and Ross, M. (2013). *The Data Warehouse Toolkit : The Definitive Guide to Dimensional Modeling*. Wiley, Indianapolis, IN, 3 edition.

[Lemainque, 2016] Lemainque, F. (2016). *Travaux pratiques avec Excel 2016 : Saisie et mise en forme, formules et exploitation des données, courbes et graphiques...* Travaux pratiques. Dunod.

[Malinowski and Zimányi, 2008] Malinowski, E. and Zimányi, E. (2008). *Advanced Data Warehouse Design : From Conventional to Spatial and Temporal Applications*. Springer Publishing Company, Incorporated, 1 edition.